INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Deep Reinforcement Learning for Investing: A Quantamental Approach for Portfolio Management

Fábio Alexandre Afonso Maltêz

Master's in Data Science

Supervisor:
PhD Diana Elisabeta Aldea Mendes, Associate Professor,
Iscte - Instituto Universitário de Lisboa

October 2022

BUSINESS SCHOOL

TECNOLOGIAS E ARQUITETURA

Deep Reinforcement Learning for Investing: A Quantamental Approach for Portfolio Management

Fábio Alexandre Afonso Maltêz

Master's in Data Science

Supervisor:
PhD Diana Elisabeta Aldea Mendes, Associate Professor,
Iscte - Instituto Universitário de Lisboa

October 2022

Acknowledgments

This work is the culmination of multiple years as a university student and a special thanks to all my friends, colleagues and teachers at Iscte is in order. This great environment that I was a part of allowed me to dive deep through all subjects that I was curious about, fostering my creativity and always pushing me to learn more and be even more curious. A special thanks must be given to Professor Diana Mendes, who motivated me to join this masters program and supported me since the beginning of this work. I must say that I'm thankful for being part of a generation that has quality and free education at the length of a click. If it wasn't for free quality teachings like the Reinforcement Learning course, provided by David Silver, this manuscript would probably be much different and with less quality. This work goes to all my family, friends, colleagues and teachers that have helped me to grow as a person and professional. Always having my back, fostering my creativity and pushing me towards greater things. Thanks!

Resumo

Todos somos afetados pelo mundo dos investimentos. A forma como o excedente de capital é alocado tanto por nós como por fundos de investimentos determina a forma como comemos, inovamos e até mesmo como fornecemos educação às crianças. Gestão de portfólio é uma tarefa essencial e desafiadora neste processo (Leković, 2021). Envolve gerir um conjunto de ativos financeiros com o objetivo de maximizar os retornos por unidade de risco, tendo em consideração todas as relações complexas entre fatores macro e microeconómicos, sociais, políticos e ambientais.

Este estudo pretende avaliar de que forma a técnica de *machine learning* intitulada de Aprendizagem por Reforço Profunda (ARP) consegue melhorar a tarefa de gestão de portfólios. Também tem um segundo objetivo de entender se variáveis relacionadas com a performance financeira de uma empresa (i.e., vendas, passivos, ativos, fluxos de caixa) melhoram a performance do modelo. Após o estado-de-arte ter sido definido com a revisão de literatura, utilizou-se o método CRISP-DM da seguinte forma: 1) Entendimento do negócio; 2) Entendimento dos dados; 3) Preparação dos dados – dois conjuntos de dados foram preparados, um apenas com variáveis de mercado (i.e., preço de fecho, volume transacionado) e o outro com variáveis de mercado mais variáveis de performance financeira; 4) Modelagem – usou-se os modelos *Advantage Actor-Critic (A2C), Deep Deterministic Policy Gradient (DDPG) e Twin-delayed DDPG (TD3)* em ambos os conjuntos de dados; 5) Avaliação.

Em média, os modelos apresentaram o mesmo índice *sharpe* nos dois conjuntos de dados – média de 0.35 vs 0.30 para o modelo base, no conjunto de teste. Os modelos ARP apresentaram uma melhor performance do que os modelos tradicionais de otimização de portfólios e a utilização de variáveis de performance financeira melhoraram a robustez e consistência dos modelos. Tais conclusões suportam o uso de modelos ARP e de estratégias de investimentos *quantamentais* na gestão de portfólios.

Key Words: Aprendizagem por Reforço Profunda; Investimentos; Gestão de Portfólio; Finanças Quantitativas, Estratégias de Investimentos Quantamentais

Abstract

The world of investments affects us all. The way surplus capital is allocated by ourselves or investment funds can determine how we eat, innovate and even educate kids. Portfolio management is an integral albeit challenging process in this task (Leković, 2021). It entails managing a basket of financial assets to maximize the returns per unit of risk, considering all the micro and macro economical, societal, political and environmental complex causal relations.

This study aims to evaluate how a machine learning technique called deep reinforcement learning (DRL) can improve the activity of portfolio management. It also has a second goal of understanding if financial fundamental features (i.e., revenue, debt, assets, cash flow) improve the model performance. After conducting a literature review to establish the current state-of-the-art, the CRISP-DM method was followed: 1) Business understanding; 2) Data understanding; 3) Data preparation – two datasets were prepared, one with market only features (i.e., close price, daily volume traded) and another with market plus fundamental features; 4) Modeling – Advantage Actor-Critic (A2C), Deep Deterministic Policy Gradient (DDPG) and Twin-delayed DDPG (TD3) DRL models were optimized on both datasets; 5) Evaluation.

On average, models had the same sharpe ratio performance in both datasets – average sharpe ratio of 0.35 vs 0.30 for the baseline, in the test set. DRL models outperformed traditional portfolio optimization techniques and financial fundamental features improved model robustness and consistency. Hence, supporting the use of both DRL models and *quantamental* investment strategies in portfolio management.

Key Words: Deep Reinforcement Learning; Investments; Portfolio Management; Quantitative Finance; Quantamental Investment Strategies

Table of Contents

List of Tables

List of Figures

1. Introduction

The main goals of this study can be divided into two categories: First, we assess the current state of the art reinforcement learning (RL) methods applied to portfolio management; second, we implement a RL algorithm that performs better than the traditional counterpart methods, being enhanced through the use of fundamental investment features.

Investment strategies can range from short-term to long-term approaches. Differences between these two strategies can lie not just on the holding period of the financial asset, but also in the investor's mindset and intent (Warren, 2016). For example, according to Warren (2016), long-term investing can be defined as a "fundamental, research-oriented investment approach that assesses all risks to the business and which has a focused discipline of seeking positive returns over the long-term business cycle". With this statement, the author is defending that a long-term investor should be defined by its latitude, intent, capacity for patience, trading discretion, investment approach and mindset/attitude (being latitude and intent the main ones). On the latitude side, a long-term investor must have the psychological strength to remain patient and demonstrate proper trading discretion throughout any possible market condition (especially during major turmoil). On the intent side, that investor must also demonstrate that he/she has a strategy, investment approach, philosophy, processes, motivation and mindset that favors wealth generation in the long-term.

Moving to the short-term side of investment approaches, Venkataramani and Kayal (2021) state that short-term investors' primary focus is to predict the market price changes of the corresponding financial assets. Thus, their decisions are inherently more dependent on daily price fluctuations. The authors also mention that "share prices trace a random walk and are difficult to predict". One of the most useful short-term strategies is market timing, which heavily relies on predicting those challenging to predict prices as precisely as possible. Venkataramani and Kayal (2021) also mention that short-term market timing strategies can produce significant returns. Still, the investor needs to possess the required capabilities and resources to leverage these strategies – explaining why high-frequency traders can gain a lot with these strategies, while retail investors don't.

Independently of the type of strategy, when the investment activity starts to entail managing a basket of financial assets, it is now entering the realm of portfolio management. Portfolio management is the process of selecting and managing a group of financial instruments - such as stocks, bonds and derivative instruments – aiming at maximizing the return on

investment while minimizing the risk/volatility (Soleymani & Paquet, 2021). Modern portfolio theory suggests that risk-averse investors maximize their wealth while minimizing risk (Venkataramani & Kayal, 2021). Considering that investors suffer biases such as: representativeness, anchoring, availability bias, overconfidence, mental accounting and herd behaviour, it is essential to formulate investment strategies that help override these human biases (Venkataramani & Kayal, 2021).

According to Jiang and Liang (2017), traditional portfolio management methods can be classified into four classes: "Follow-the-Winner", "Follow-the-Loser", "Pattern-Matching" and "Meta-Learning". These are either based on prior-constructed financial models, using historical patterns under some assumptions on market behaviour, or a combination of various models (Jiang & Liang, 2017).

If we now look into the historical evolution of portfolio management, Lekovic (2021) provides a framework that splits this evolution into three main phases: traditional portfolio theory (TPT), modern portfolio theory (MPT), and post-modern portfolio theory (PMPT).

Traditional portfolio theory started to appear at the beginning of the 20th century, and the investor's main focus was performing fundamental analysis of the securities in the portfolio. During this time, investors were only starting to apply scientific methods to their research process through the study of the company's financial statements – this was also the period when stricter financial statements' control was put into place for companies that were listed in the stock exchange. The other tool that investors used was naïve (simple) diversification, where an increase in the number of securities in the portfolio was believed to lead to a decreased risk level of the same portfolio (Leković, 2021).

Then it came Modern Portfolio Theory. "MPT provides a mathematical framework for optimizing return and risk ratio, and goes a step further than TPT, since the focus shifts from the analysis of individual securities to the analysis of portfolio characteristics." (Leković, 2021). Harry M. Markowitz is one of the main creators of MTP, and this theory states that return is a risk function that can be reduced by efficient diversification (through a low correlation between returns of portfolio securities) – not naïve diversification, as it was in the case of TPT. This was the first time that the trade-off between risk and return was formally quantified. With MTP came many useful portfolio management tools such as the sharpe ratio, efficient frontier, CAPM (Capital Asset Pricing Model) and ATP (Arbitrage Pricing Theory). But it still had some shortcomings that needed to be addressed: it is a static analysis (there are no adjustments to the portfolio after the initial decision); the possibility of buying/selling securities in unlimited proportions; conversion of correlation coefficients to one during financial crises; assumption

that investors have homogeneous expectations; variance was believed to be a reliable risk measure; and assuming that the returns on financial assets follow a normal distribution (Leković, 2021).

The last phase is Post-Modern Portfolio Theory. PMPT was built on top of MPT, as a way to try to fix its errors and provide better tools to construct optimal portfolios for each investor. The main differences are the way that PMPT interprets risk, and treats positive and negative variance of returns differently. According to PMPT, each investor has an individual minimum acceptable return (MAR), which can be interpreted as the investor's target return that can be used as a benchmark when real performance is being evaluated. "Unlike MPT that associates risk with achieving an average return, PMPT claims that the investment risk should be linked to the specific objective of each investor, and that returns above this objective do not represent an economic or financial risk. According to PMPT, only volatility below the investor target return is considered risk. Return above the target creates uncertainty, which is nothing but a risk-free opportunity to achieve unexpectedly high return" (Leković, 2021).

Due to technological advancements that allowed for the massive availability of data and the surge of tools to analyze it, big data analytics and machine learning fields started to emerge and progress rapidly (Hu & Lin, 2019). And as these fields tackle complex problems, the use of machine learning and, more specifically, deep reinforcement learning for portfolio optimization problems started to be applied and studied by the scientific community (i.e., Aboussalah & Lee, 2020; Betancourt & Chen, 2021; Ren *et al.*, 2021; Tsantekidis *et al.*, 2021).

Aboussalah and Lee (2020), as an example case, refer to this type of new strategy as "automated data-driven investments", relying on machine learning agents that through consistent and systematic trading techniques, provide an alternative to more traditional trading strategies developed on the bases of microeconomic theories. Some major flaw of these conventional techniques and even of some machine learning algorithms is that they need to create either rigid assumptions (i.e., returns follow a normal distribution) or simplify the world in which the trading agent is trained (i.e., limiting the number of actions that the trader can take at a specific time period). Meaning that when the algorithm is tested in the real environment, its performance is subpar due to the higher level of complexities that the model/algorithm/agent must face in the real world. And after the authors acknowledged this issue, they proposed a new reinforcement learning algorithm that can improve the agents' performance level. For this specific case, the new algorithm presented was called Stacked Deep Dynamic Recurrent Reinforcement Learning (SDDRRL), and solved the major constraints shown above: it can perform multiple continuous actions for a diverse set of assets whilst abiding the portfolio

constraints – this also allows the model to perform better in noisy environments; and since it uses deep learning, there's no need to formulate strong assumptions. This is just one example of how an advanced machine learning model, coupled with a good data set, could better tackle the issue of optimizing a financial portfolio.

Moreover, machine learning techniques, such as Deep Neural Networks (i.e., LSTMs), that have been employed in trading strategies, rely on the prediction of prices that are seemingly impossible to truly predict timely and accurately (Sun *et al.*, 2021; Zhang *et al.*, 2021). With Deep Reinforcement Learning (DRL), the process of predicting values can be forgotten and the focus of the model can be directly on how to allocate the portfolio assets in the best way possible (Gu *et al.*, 2021). Reinforcement Learning (RL) and Deep Learning (DL) have characteristics that complement each other – RL has a good capacity for decision-making but has major drawbacks in perception, while DL is strong in perception and has weak decision-making capacity (Khemlichi *et al.*, 2020). Thus, DRL can be used to solve the problem of optimal decision-making in complex environments (Khemlichi *et al.*, 2020).

Overall, the increasing levels of available data and the continuous improvements in machine learning models were also highly felt in the investment world, mainly on the quantitative finance side. Yet another example of the benefits these models bring to investments is presented by Spiegeleer *et al.* (2018), where they promote a machine learning framework that increases by many folds the speed with which it can calculate the prices of complex financial assets. The algorithm is called Gaussian Process Regression (GPR), and through the parametrization of the financial assets' main characteristics, it is able to price options way faster in comparison to more traditional and time-consuming techniques (i.e., Monte Carlo Simulations). It's worth noting that with the speed increase (and efficiency gains) comes some loss in accuracy, but Spiegeleer *et al.* (2018) comment on this issue with the following statement: "The price we have to pay for this extra speed is some loss of accuracy. However, we show that this reduced accuracy is often well within reasonable limits and hence very acceptable from a practical point of view".

The main goal of this study is to analyze the performance of three Deep Reinforcement Learning algorithms on the task of financial portfolio management and compare it against benchmarks. Importance is also given to the type of data being used. Throughout the literature used for this study, only one author notably mentioned that the limited data and features used were a drawback to the machine learning model's performance (Kang *et al.*, 2018). Therefore, this study will also emphasize on how important good features regarding a portfolio company's fundamental performance (not just market related features) can be. This will be achieved by

comparing the performance of these models with just market features and then with market plus fundamental features.

The complex world of investments affects all of us directly or indirectly. From the bank deposit we make to save for our kids' college, to the investment funds that invest in growing companies and even to how government allocates taxpayer money to build public infrastructure – all these tasks can relate to portfolio management. The importance of this study relies on two main pillars: i) it reinforces the overall benefits that DRL algorithms bring to the portfolio management activity; ii) it emphasizes the importance of using features related to the portfolio companies' fundamentals – features related to the companies' financial performance, such as profit, debt, cash, etc. Hence, this study will also try to bridge quantitative and fundamental investment strategies, which entails venturing into the new term of *quantamental* investment strategies.

The remainder of this dissertation is structured in the following manner: theoretical background on portfolio management, quantitative finance and reinforcement learning; literature review of deep reinforcement learning theory; the methodology used in training and testing the DRL models; presentation of results and discussion of them; and it finishes with a conclusion of the main findings and study contributions, as well as major limitations and future research.

2. Theoretical Background

2.1 Portfolio Management

Financial Portfolio Management is the activity of managing a basket of financial assets, whilst trying to achieve the goal set by the investor – which usually entails maximizing returns and minimizing risk (Soleymani & Paquet, 2021). This activity can be recognized as one of the main tasks of financial experts worldwide (Leković, 2021). The remainder of this section will briefly explain the fundamental portfolio management topics and concepts needed to understand the analysis and results presented in this study.

i. Modern Portfolio Theory

Modern Portfolio Theory (MPT), developed by Markowitz (1952) and known as mean-variance analysis, consists of a mathematical model that improves financial theory by providing an objective and systematic approach that enables establishing and optimizing the relation between expected return and assumed risk (dos Santos & Brandi, 2017; Leković, 2021). Putting MPT on a historical time series, it comes in the middle between Traditional Portfolio Theory and Post-Modern Portfolio Theory, and it can be considered a major breakthrough as it was the first time that investors had a scientific, objective and quantitative tool that allowed them to look at their portfolios as a whole – including all the important details that come with it, such as controlling the covariance between returns of portfolio companies (Leković, 2021).

Such was the importance of MPT, that we still see it being considered by the scientific community as a valid option when managing large pools of capital. One example of this is provided by Lord (2020), as it uses MPT principles as the bases for the decision-making process of university endowments. Despite the fact that the main goal of Lord's article was to prove that an experienced, diverse and open-minded investment committee contributed to a more diverse and less risky fund, the author uses the MPT framework as an "intermediary of the effect of committee characteristics and norms on portfolio diversification" – in other words, the author proposes using the objective Modern Portfolio Theory as a way to offset the personal and group behavioral biases that the committee might demonstrate during the decision making process.

ii. Rate of Return

Rate of return is the profit generated by an investment, as a percentage amount of the initial capital invested (Verdiyanto *et al.*, 2020). The rate of return, r_t, can be defined as follows, being P_t the closing price at time t:

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

iii. Volatility: Variance and Standard Deviation

According to Verdiyanto (2020), "variance (σ_i^2) corresponds to a mathematical calculation in a data collection of the spread between numbers". The variance of asset i is determined by the following equation:

$$\sigma_i^2 = Var(r^i) = \frac{\sum_{t=1}^{m}(r_t^i - \mu_i)^2}{m-1}$$

Where r_t^i is the value of asset i at time t, μ_i is the mean value of asset i across the sampled timeframe and m is the number of periods in the sample (i.e., 360 days).

The standard deviation is a measure of dispersion in comparison to its mean, and is measured as the variance's square root (Verdiyanto *et al.*, 2020):

$$\sigma_i = \sqrt{\sigma_i^2}$$

iv. Sharpe Ratio

Sharpe ratio is a portfolio performance metric that measures the excess return, per unit of risk. Here, risk is determined by the standard deviation (σ_p) and excess return is the return minus the risk-free rate (r_f) (Verdiyanto *et al.*, 2020). Sharpe ratio (S_p) can be expressed as:

$$S_p = \frac{E\,(r_p - r_f)}{\sigma_p}$$

v. Efficient Frontier

The efficient frontier is a set of ideal portfolios that allow an investor to have the highest expected return for a certain level of risk, or the minimum level of risk for an arbitrary expected

return level. This is a useful tool for the investor to focus the decision-making process on the trade-off between risk and expected returns (Verdiyanto *et al.*, 2020).

2.2 Quantitative Finance

Quantitative finance is the field that uses quantitative and computerized data-driven models to help investors and traders to make better investment decisions. These quantitative models can be considered an extension of human capabilities that serve as intermediaries between traders and markets - and as technology further advances and data availability increases, these models are becoming ever more complex and sophisticated. Machine Learning can be considered a subsector of quantitative finance, and as its use by top-tier hedge funds and trading firms is increasing, it is believed that in the coming years machine learning will change even more the way people trade and invest in the financial markets (Hansen, 2021).

This chapter subsection will go deeper into explaining two important quantitative finance-related concepts for this study: sequential least squares programing method and backtesting.

i. Sequential Least Squares

Sequential Least Squares is an optimization method to solve nonlinear optimization problems. Fu *et al.* (2019) explains it as a two-step method. First, one needs to identify a nonlinear least squares problem, and second, the problem must be transformed into a sequential quadratic programming model so that it can be solved. The following terminologies and mathematical explanations are based on this article.

In broad terms, a sequential quadratic programming model can be framed as follows:

$$\min f(x) = \frac{1}{2}\, x^T G x + \; g^T x$$

$$s.t. \quad Ax = b, \qquad x \geq 0$$

Where $G \in R_{\mathrm{nxn}}$ is a symmetric matrix set to its Hessian matrix $H(f(x_i))$, $g \in \mathrm{R}_{\mathrm{nx1}}$ is a gradient vector $\nabla\, f(x_i)$, $A \in R_{mxn}$ and $b \in R_{mx1}$ belong to the optimality constraint stating Ax vector must be equal to b vector.

This optimization problem is optimized iteratively, meaning that it is divided into sequential subproblems. Once one optimal point in a subproblem is found, it will search for the

next feasible point starting from the current optimal point. If d_k is the solution of the optimal subproblem, then:

$$x^{(k+1)} = x^k + \alpha_k d_k$$

Where $\alpha_k \in [0, 1]$.

A nonlinear problem is defined as follows:

$$L + \Delta = f(x)$$

Where L is a (nx1) vector, f is a nonlinear function, and Δ is a (nx1) observational error vector. On this nonlinear model, the following error equation also has to hold:

$$V(x) = f(x) - L$$

To finalize, the above nonlinear model can be converted into a quadratic optimization problem, where $F(\Delta x^k)$ is the nonlinear function to be iteratively optimized, as follows:

$$\min \quad F(\Delta x^k) = V^T P V$$

$$s.t. \quad V = f(x^k) - L$$

ii. Backtesting

In its simplest form, a backtest is a simulation of how a model would hypothetically have performed during a historical time period (López de Prado, 2018, p. 151).

In his book "Advances in Financial Machine Learning", López de Prado (2018) explains why backtesting is important and, at the same time, why it is a complex and usually misused concept in quantitative finance. From this book, one can comprehend that a solid backtest is essential to understand whether a model's final version is proper. It appears to be common for the scientific community and investment professionals to think of backtesting as a tool to improve a model during the iterative research process. This leads to model overfitting in the backtest and false discoveries that underperform when implemented in the real world.

The author goes deeper into this subject and states that backtest is not a research tool, but if used correctly it can be an important tool to do a sanity check and understand that the model is helpful under certain constraints and market scenarios. Since the historical time period chosen for the backtest is arbitrary, it will never happen again in the future. It's not a good

practice to think that if we run the backtest multiple times with the same model on the same dataset, that the possible causal relations found between features and model performance are true discoveries – often is the case that these are false discoveries caused by backtest overfitting. The best way to avoid this pitfall is to backtest only the final version of the model, and if the backtest is not good we should redo the model again. López de Prado puts it in the following manner, calling it the *Marco's Second Law of Backtesting*: "Backtesting while researching is like drinking and driving. Do not research under the influence of a backtest".

2.3 Reinforcement Learning

Reinforcement learning (RL) is a semi-supervised ML algorithm where an agent tries to act optimally throughout a sequential decision-making process, given a goal and an environment state (*figure 1*) – having a lot of applications in the investment world. The main differences that RL has compared to supervised and unsupervised machine learning is that it has a partial feedback loop, and that the RL agent must balance between exploring the environment with new actions or exploiting through using the actions it already knows will output high levels of reward (*exploration – exploitation dilemma*). Regarding the first difference, when taking actions at a certain state, the agent receives a reward from the environment - which are quantitative values but not explicit regarding the action being right or wrong, it only outputs a value. This reward is a partial feedback because the agent never knows if that's the highest reward possible, but on the other side it has the goal of maximizing the total cumulative reward over a sequence of steps. This feedback process creates a loop because when the agent takes an action from a certain state, a reward is generated and the environment state changes in accordance with the action (sometimes also due to some environment stochasticity), and the same happens in the next state-action-reward-state iteration. The second difference is that the agent never knows if it achieved the highest reward possible, leading to the need to randomly explore new actions to maximize total cumulative returns. But on the other hand, the agent already knows which actions produce the highest returns until that moment, and not reselecting them might represent missing an exploitation opportunity – therefore, the RL agent is faced with the exploration-exploitation dilemma (Dixon *et al.*, 2020, Chapter 9).

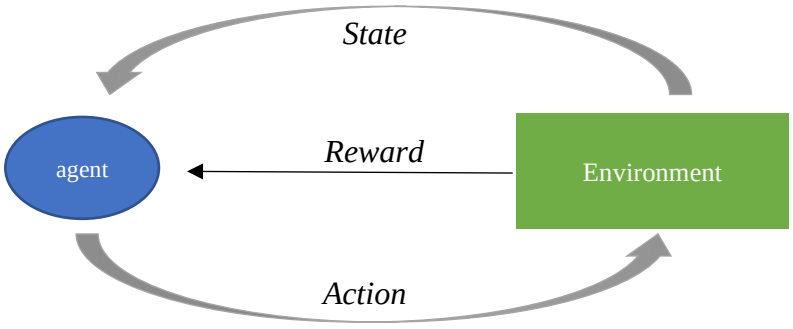

Figure 1: Agent - Environment Relation

The remainder of this chapter will primarily focus on using Dixon *et al.*'s (2020) book to introduce the reader to the basic reinforcement learning concepts, which will be essential to understanding the rest of this study. The following concepts will be explained: rewards; value and policy functions; observable and partially observable environments; Markov decision processes; and Bellman equations.

i. Rewards

The reward function is used to determine the goal of the reinforcement learning problem. To be more precise, the agent's goal will be to maximize the total cumulative reward over a certain period. Rewards are quantitative measures of the level of satisfaction/dissatisfaction the agent gets when it acts in a state (Dixon *et al.*, 2020, p. 284). The local reward immediately received after an action can be mathematically written as $r_t = r_t(S_t, a_t)$, where the reward r_t depends on state S_t, action a_t and time period t (in the case of a time-dependent problem).

Usually, the RL problem will require optimizing total cumulative rewards over T steps, where rewards and actions taken at one time step will impact the environment and the actions taken in the subsequent steps (rewards over multiple time steps are not independent) – which relates to the feedback loops mentioned above.

ii. Value and Policy Functions

We already know that rewards depend on the action taken (a_t) and the current environment state (S_t). Different states in the environment can have different levels of attractiveness / value to the RL agent – i.e., state x can output low rewards independently of the action taken, not being an interesting state for the agent to be at. *Value function* is the RL concept used as a numerical method for the agent to access the level of attractiveness of state

S_t. The value function can be defined as "a mean (expected) cumulative reward, that can be obtained by starting from this state, over the whole period" (Dixon *et al.*, 2020, p. 286).

To determine the value function, one needs to know beforehand how the agent will behave first, because the rewards depend on both the state and the action taken. The rule that states how the agent should act in any possible state is the *policy function* π_t (S_t). It can be deterministic (deterministic function of the state S_t) or stochastic (probability distribution over a range of possible actions). Hence, we have the value function V^{π} (S_t) that depends on the current environment state S_t and policy π.

iii. Observable and Partially Observable Environments

Let's go deeper into the notion of a state. For an agent to take an action, it needs to understand the state it is in, and after the action is taken the agent will be in a new state provided by the environment. This process of the agent comprehending the environment and acting upon the information it has is extended over multiple periods of time. So, how much information does the agent need to make an informed decision? For example, if the agent is a robot walking in a maze, it's impossible to know the whole environment but it still needs to make a decision and act.

In the example provided above, the agent only gets to partially observe the environment (the robot can't see the full maze because it is inside of the maze). And therefore, it is important to understand if the agent can fully or only partially observe the environment in the current state. So, a fully observable environment is when the agent can see the entire environment (i.e., if the robot agent could see the whole maze during all states and all time steps); and a partially observable environment occurs when the agent can't see the whole "picture" of the environment and has to take action with sub-optimal information (i.e., what the robot sees when walking in the maze, having to take decisions nonetheless).

A way to simplify most of RL problems is to assume that they follow Markovian dynamics. According to Dixon *et al.* (2020), these dynamics assume that the transition probability $p(s_t \mid s_{0:t-1})$, at time t, of the conditional state s_t, depend not on the full history but rather only on the k most recent values. And if $k = 1$, which is the most common case (Dixon *et al.*, 2020, p. 287), we get the following:

$$p\,(s_t \mid s_0, s_1, \dots, s_{T-1}) = p\,(s_t \mid s_{T-1})$$

iv. Markov Decision Processes

"Markov Decision Processes are a tool for modeling sequential decision-making problems where a decision maker interacts with a system in a sequential fashion" (Szepesvári, 2009, p. 8).

Extended from the Markov model, the Markov Decision Process (MDP, figure 2) provides new degrees of freedom that consist of the agent's actions. With these actions, the RL problem gains control variables that have impact into the feedback loop of the Markov process. The MDP framework allows one to describe the goal-oriented learning process through the multiple agent-environment interactions. Mathematically, it is described by a set of discrete time steps $t_0, \dots, t_n$ and a tuple $\{S, A(s), p(s' \mid s, a), R, \gamma\}$. In this tuple, we have the following elements, respectively: states S; the set of actions $A(s)$ that can be taken at step s; the transition probability of state s', knowing that at state s the action a was taken; the rewards function R; and finally the discount factor γ, which is a number between 0 and 1 used to discount future rewards when calculating the value (total cumulative rewards) of a state.

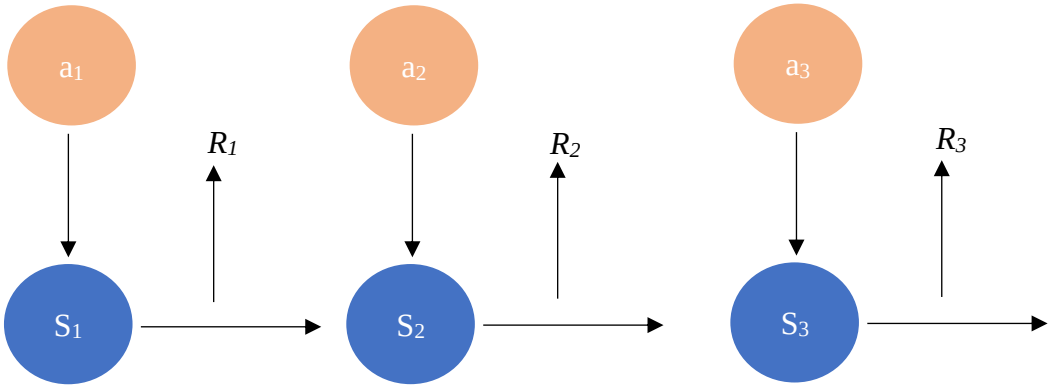

Figure 2: MDP Framework

v. Bellman Equations

We already know that the state-value function relies on knowing the total cumulative reward of each step, also known as "value". In reinforcement learning, the random cumulative rewards G_t (that vary according to the policy), can be defined as follows:

$$G_t = \sum_{i=0}^{T-t-1} \gamma^i \, R\,(S_{t+i}, a_{t+i}, S_{t+i+1})$$

While defining the state-value function, it was noted that it only partially described the rewards' dynamics because these depend on the actions. Therefore, we can define an action-value function $Q^{\pi}(s, a)$ that specifies the value according to the state and action, while following policy π. The following expression defines the action-value function:

$$Q_t^{\pi}(s, a) = E^{\pi}[G_t \mid S_t = s, a_t = a]$$

These value functions can also be represented by a simple recursive scheme that computes the value function at time t in terms of its future values at time $t + 1$ by going backward in time (Dixon *et al.*, 2020, p. 295). These recursive relations are known as the "Bellman equations", a key concept that underpins RL frameworks.

Given the equation for the action-value function as an example, this recursive relation can be formulated as follows:

$$Q_t^{\pi}(s, a) = E_t^{\pi}[R_t(s, a, s')] + \gamma E_t^{\pi}[V_{t+1}^{\pi}(s')]$$

Where $Q_t^{\pi}(s, a)$ is the action-function under policy π, at time t. $E_t^{\pi}[R_t(s, a, s')]$ is the expected reward of choosing action a, in state s, and ending up in state s' - under policy π, at time t. And $\gamma E_t^{\pi}[V_{t+1}^{\pi}(s')]$ is the expected total value of state s', at time $t + 1$, under policy π and discounted by γ.

3. Literature Review

Now that the reader is acquainted with the main terminologies that connect investments with reinforcement learning, this new chapter will delve into the concrete problem tackled in this study. This literature review will properly define the main research question and the actual literature review will be conducted; it ends with justifying the deep reinforcement learning algorithms used in this study, and explaining how they work in broad terms. This is an important step to grasp the state-of-the-art methods currently being applied to solve the problem at hand, and to understand how this study can contribute to the scientific community.

As it was already mentioned, the goal of this study is to access the importance that DRL techniques have in portfolio management and understand if features related to the company's financial fundamentals have a positive impact in the model's performance. Therefore, the following research question was proposed: *Can deep reinforcement learning improve medium-long term fundamental investment techniques, in a portfolio of stocks?*

After the research question got defined, the following steps were taken to perform the literature review: select database and keywords; material extraction and selection; analysis of results. The reader needs to understand that the articles used in this literature review are not the only ones mentioned in this study, nor are they the only ones about the subject. This literature review consists solely of a structured and consistent approach to justify the importance of the research question as well as the DRL models used in this study.

Scopus was the database used, and the query was: *(Deep reinforcement learning OR reinforcement learning) AND (investments OR portfolio management OR quantamental investments)*. It returned a total of 44 results. After a first screening based on the title, keywords and abstract, 33 articles remained. Finally, after reading the selected articles, 24 were chosen to be the main articles discussed. The analysis of results will be based on the 33 articles, due to the quantitative nature of the analysis and the fact that all the articles are within the selected query. While in the final discussion of this literature review step, 24 articles will be used because these focused on using deep reinforcement learning for portfolio management.

3.1 Literature Review Analysis

The first observation that can be made with this study is that DRL for portfolio management is a recent research topic, with the oldest articles from 2018 – one that used DRL for the stock market (Kang *et al.*, 2018) and another one for the cryptocurrency market (Jiang & Liang, 2017).

Year	Citations	Articles
2018	65	4
2019	5	3
2020	46	12
2021	9	14
Total	**125**	**33**

Table 1: Number of articles and citations, per year

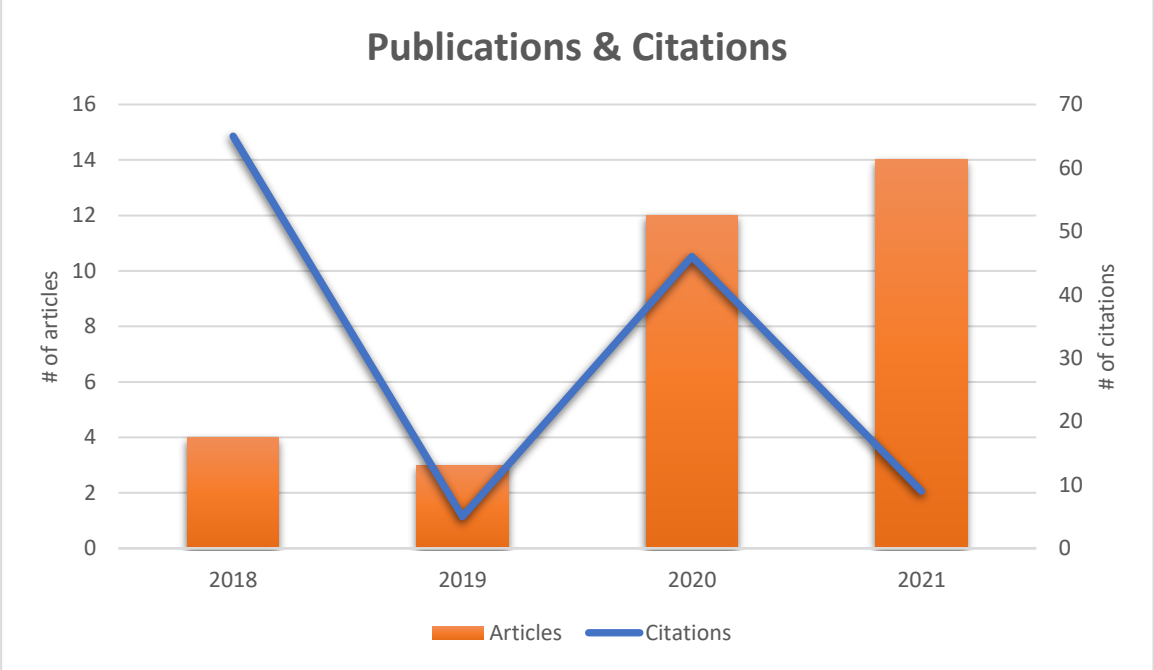

Figure 3: Number of articles and citations, per year

From the table and figure above, it can be seen an evident growth in research publications about reinforcement learning for portfolio management. In 2020, attention to this research problem grew and multiple DRL approaches started to be applied to investment portfolios of different kinds – ranging from portfolios of stocks, to portfolios of cryptocurrencies and even to more complex portfolios with multiples types of financial assets (i.e., Gao *et al.*, 2020; Lin *et al.*, 2020; Lucarelli & Borrotti, 2020). The peaks of citations observed in 2018 and 2020 happened due to, respectively, an innovative study on portfolio management of cryptocurrencies and a DRL methods that, according to its authors, outperforms more "traditional" DRL methods in noisy environments such as the one of financial markets (Aboussalah & Lee, 2020; Jiang & Liang, 2017).

Year	Article	Book	Conference Paper	Total
2018			4	**4**
2019			3	**3**
2020	5	1	6	**12**
2021	6		8	**14**
Total	**11**	**1**	**21**	**33**

Table 2: Types of research publications, per year

Regarding types of research publications, conference papers were the most used for this study (21), followed by journal articles (11).

It's also worth mentioning that, in 2021, more challenging problems related to portfolio management started to be tackled. In the case of the articles selected, one can begin to see the application of deep reinforcement learning methods to solve problems related to financial hedging strategies and to portfolio management in the insurance industry (where there are more constraints to be considered) (Abrate *et al.*, 2021; Pham *et al.*, 2021).

In terms of results, most authors focused on using Sharpe Ratio as a performance metric (i.e., Huang *et al.*, 2021; Sun *et al.*, 2021; Zhang *et al.*, 2021). In contrast, Gu *et al.* (2021) focused simply on the compounded return of the investments.

It was already mentioned that portfolio management considers both the return and risk of assets. And as stated by Shi *et al.* (2019), Sharpe Ratio is a "risk adjusted mean return", therefore, one could say that this is a more appropriate measure of the performance and success of the algorithm applied.

Besides the different measures of performance used, the fact that other baseline methods and investment periods are considered makes the comparison of results that much more difficult. Kang *et al.*'s (2018) main baseline was the performance against the S&P 500 index, while Shi *et al.* (2019) used as benchmark the performance of other traditional and machine-learning based models. Betancourt *et al.* (2021) used 11 months' worth of data in their test set, while Shi *et al.* (2019) used two months' worth of data in their test set.

Nevertheless, if we focus on the studies that had traditional models as benchmarks included. Gao *et al.*'s (2020) Deep Q-Network obtained a Sharpe Ratio of 23.07% - twice as much as the 2nd and 3rd best traditional models. Zang *et al.*'s (2021) DDPG model improved Sharpe Ratio in 33% against benchmark. And the innovative Ensemble of Identical Independent Evaluators approach, by Sun *et al.* (2021), improved Sharpe Ratio by at least 50%, compared to the benchmark.

Portfolio management and optimization through deep reinforcement learning is a growing hot topic for researchers (i.e., GAO *et al*., 2021; Huang *et al*., 2021; Soleymani & Paquet, 2020). The DRL algorithms that attract researchers the most are actor-critic (more specifically, the Deep Deterministic Policy Gradient method, DDPG) and Q-Learning models (i.e., Gao *et al*., 2020; Khemlichi *et al*., 2020; Lucarelli & Borrotti, 2020; Zhang *et al*., 2021). On the other side, some authors tried to use more innovative RL frameworks that are believed to have better performance when facing complex and noisy environments (i.e Gu *et al*., 2021; Lee *et al*., 2020; Shi *et al*., 2019).

Despite the novelty applied to new algorithms with implementation in various types of investment portfolios, there is a major limitation on the type of features used – a constraint mentioned by Kang *et al.* (2018). Practically all articles for this systematic literature review used only market related features – either using the price, volume and/or other features derived from these two (i.e., Harnpadungkij *et al*., 2019; Ren *et al*., 2021; Xu & Tan, 2020).

3.2 DRL Algorithms

As stated in the literature review, actor-critic deep reinforcement learning algorithms have been one of the most used methods to solve portfolio optimization problems with RL. Therefore, this study focuses on using and analyzing the performance of three actor-critic algorithms: A2C (Advantage Actor-Critic), DDPG (Deep Deterministic Policy Gradient) and TD3 (Twin-delayed DDPG).

Actor-critic methods generate both a policy and value function. Here, the *actor* is the algorithm that generates the policy function from the family $\pi_\theta\ (a|x)$, and the *critic* evaluates the results outputted by the actor, expressing it as a state-value or action-value function (Dixon *et al*., 2020, p. 310).

i. Advantage Actor-Critic (A2C)

A2C maintains a policy and an estimate of the value function (Mnih *et al*., 2016). In this method, the critic learns the value function, and the actor network learns the policy in the direction set by the critic (Park & Lee, 2021). In broad terms, the loss function of the critic network (V_θ) is defined as follows:

$$L_{critic} = E[(r + \gamma V_\theta(s') - V_\theta(s))^2]$$

And the loss function of the actor network (π_φ) is defined as follows:

$$L_{actor} = E[-\log \pi_\varphi(a|s)(r + \gamma V_\theta(s') - V_\theta(s))]$$

Hence, the critic network updates are based on the Bellman equation and the actor network updates are based on stochastic policy gradient theory (Park & Lee, 2021). Furthermore, the A2C critic network also estimates the advantage of an action a in state s, $A(s, a)$:

$$A(s, a) = Q(s, a) - V(s)$$

ii. Deep Deterministic Policy Gradient (DDPG)

Park and Lee (2021) provide a good explanation of the DDPG framework. This actor-critic method also combines Deterministic Policy Gradient (DPG) and Deterministic Q-Network (DQN) methods. In DDPG, noise (N) is added to the policy:

$$a = \mu_\varphi(s) + N$$

The loss function of the critic network, which uses target Y, is defined as follows:

$$L_{critic} = E[(Y - Q_\theta(s, a))^2]$$

where,

$$Y = r + \gamma Q_{\theta'}(s', \mu_{\varphi'}(s'))$$

The loss function of the actor network, which uses the deterministic policy gradient theorem (Park & Lee, 2021), is defined as:

$$L_{actor} = E[-Q_\theta(s, \mu_\varphi(s))]$$

DDPG has online and target networks – the main goal is to update the target networks (Park & Lee, 2021). The above loss functions are used to update the online actor and critic networks. The target actor (θ') and critic (φ') networks are soft updated from the online networks:

$$\theta' \leftarrow \tau\theta + (1 - \tau)\theta', \quad \varphi' \leftarrow \tau\varphi + (1 - \tau)\varphi'$$

iii. Twin-delayed DDPG (TD3)

TD3 is an improved version of DDPG, enhancing its capabilities in three ways. First, it adds Gaussian noise to the target action. This technique is known as smoothing and serves as a way to reduce overfitting to the sharp peaks produced by the Q-value estimates. Secondly, TD3 uses two critics to solve a common problem of overestimation in the DDPG algorithm – the idea is to use the minimum value in the pair of critic networks to compute the target value. Third, TD3 updates the actor network μ_{φ} less frequently than the critic network Q_{θ}, aiming to improve Q-values convergence. Therefore, TD3 improves over DDPG through target policy smoothing, clipped double Q-learning and delayed policy updates (Park & Lee, 2021).

4. Methodology

Once the goals were set, the research question defined and the literature review to back this study properly done, the experimentation phase started. Here, the focus was on the following: *i*) Does DRL performs better than traditional quantitative models? *ii*) Do features related to the portfolio companies' financial performance help improve model performance?

Specifying the importance of each of the two questions defined, if *i*) stands true we prove (in line with other studies mentioned in the literature review) that reinforcement learning has benefits to portfolio management. If *ii*) also verifies to be true, this study brings to light the importance of the field of *quantamental* investing – which brings together strategies from quantitative finance and fundamental investing, already mentioned by some authors (López de Prado, 2018, p. 53).

As this was an experiment conducted in the field of data science, the CRISP-DM methodology was applied (figure 4). Therefore, it followed the following steps: 1) Business understanding; 2) Data understanding; 3) Data preparation; 4) Modeling; 5) Evaluation – this last step will be thoroughly analyzed in the "Results and Discussion" chapter.

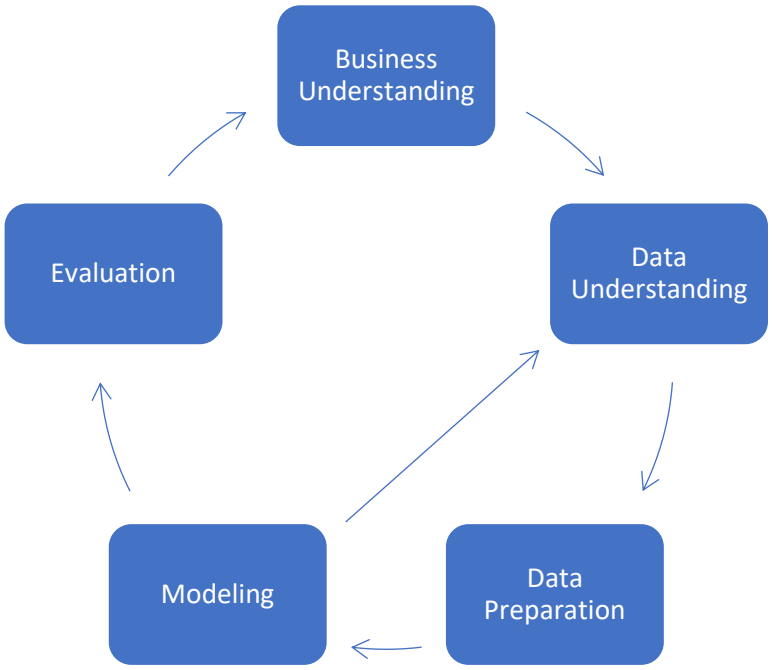

Figure 4: Experimentation Phases

4.1 Business Understanding and Data Understanding/Preparation

Regarding business understanding, it's clear by now that we want to see if DRL in portfolio management has good performance and if financial performance features help achieve a better model performance. For data understanding, it is important to remind ourselves that there will be two primary datasets going into each DRL model - one has only market features and the other has both market and financial performance features. Market data comes from Yahoo Finance python API and financial performance data comes from the Bloomberg terminal. All data used is from the beginning of 2004 until the end of 2020.

To better analyze the impact of financial performance features in the DRL model, the portfolio companies used for this study are all from the same industry. Therefore, the long-only portfolio optimized in this study is comprised of three stocks of companies from the automotive industry. These are: Peugeot; Volkswagen; and Volvo.

The data preparation phase was where the data cleaning and feature engineering steps took place. The foremost cleaning step was to ensure that prices were available for all three companies on all days. If, for a certain day, any of the three needed prices weren't available (i.e., due to holidays in a certain market), this day was removed from the analysis. Feature engineering was mainly done through the *FeatureEngineer* method of the *FinRL* python package – this step was used to add technical indicators and a volatility index to the datasets (figure 5).

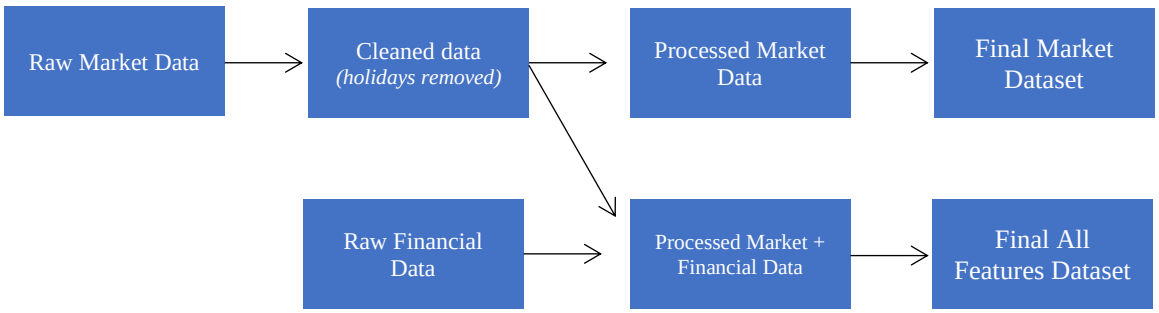

Figure 5: Data Preparation Process

The final market features dataset is a (11.739 x 20) matrix, and the full features dataset is a (10.986 x 26) matrix (table 3). This difference of 753 instances (6.4% of total instances in the market feature dataset) occurred because there were a lot of missing financial values in 2004, therefore, these instances were removed ensuring that the full features dataset started in January 2005.

The market features dataset has the following variables: *date; open daily price; high; low; close daily price; adjusted close price; daily volume traded; ticker; weekday; macd (moving average convergence divergence indicator); Bollinger upper band; Bollinger lower band; RSI 30 (Relative Strength Index for 30 days); CCI 30 (Commodity Channel Index for 30 days); DX 30 (Directional Index for 30 days); Close SMA 30 (Simple Moving Average for 30 days); Close SMA 60 (Simple Moving Average for 60 days);* and *VIX (volatility index).*

The full features dataset uses the features mentioned above, plus the following: *daily volatility; EBITDA margin; Current ratio; Assets / Equity ratio; Debt / Equity ratio;* and *ROIC (Return on Invested Capital).*

Both datasets have also two final features: a covariance matrix and a daily returns list. Financial features change every half-year, due to the periodicity with which they were taken from Bloomberg.

	Market Only Dataset	**All Features Dataset**
Shape	11.739 instances x 20 features	10.986 instances x 26 features
Start Date	January 1st, 2004	January 1st, 2005
End Date	December 31st, 2020	December 31st, 2020

Table 3: Dataset Description

For the training and test dataset, the 13th of June of 2018 was chosen as the split date to obtain approximately 85% of training data and 15% of test data. The two datasets got the following train and test sets:

	Market Only Dataset	**All Features Dataset**
Training dataset shape	9.885 instances x 20 features	9.132 instances x 26 features
Test dataset shape	1.854 instances x 20 features	1.854 instances x 26 features

Table 4: Test & Training sets description

4.2 Modeling and Evaluation

After the data preparation, the following step is to build the DRL models. In this modeling stage, a baseline model was set up and three DRL models for both datasets were built.

The baseline model used is the Sequential Least Squares Programming (SLSQP) method for portfolio optimization – the *scipy.optimize* python library was used. Here, only the closing daily prices of each portfolio company was considered. This static optimization technique allows us to create the efficient frontier for the three-stock portfolio. As we can see in figure 6, from the efficient frontier we can choose the portfolio that maximizes the return per

any given unit of risk – assuming that the portfolio companies' covariance matrix remains stable. Interesting takeaways from this analysis is that Volkswagen is the only stock that can make sense for a rational investor to hold just by itself, from a risk/return perspective, but at the same time this would also be the riskiest choice the investor could make. On the other hand, owning Volvo or Peugeot as single stocks should not be considered as there are more profitable options for the same risk these positions have.

But the two main points that one should consider from the efficient frontier are the ones that give the maximum sharpe ratio and the minimum volatility portfolios. Choosing either will depend on the investor's ultimate goal. If one is risk averse, the minimum volatility portfolio is the best choice. If one wants to maximize profit per unit of risk, he/she should opt for the maximum sharpe ratio portfolio. In this study, the latter goal was chosen, and hence, the benchmark portfolio comprised the following weights:

- Volkswagen: 62.20%
- Volvo: 31.89%
- Peugeot: 5.91%

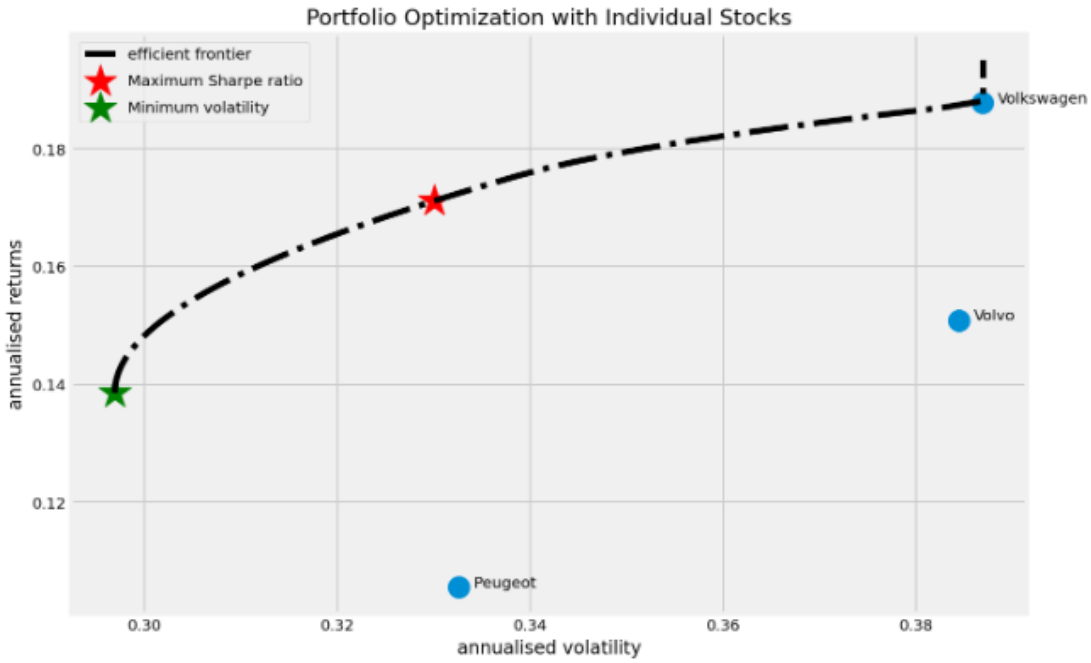

Figure 6: Efficient Frontier

The Deep Reinforcement Learning algorithms used on both datasets were the A2C, DDPG and TD3. These models were employed from the *FinRL* python package and had their respective hyperparameters tunned to optimize model performance. All these models (three plus benchmark) are explained in the "Theoretical Background" chapter.

Deep diving into the hyperparameters tunning, table 5 shows the final optimized hyperparameters of each model for both datasets. For each DRL model, the chosen hyperparameters to tune were the ones available in the *FinRL* package. For A2C, the number of steps until update and learning rate were tuned. For DDPG and TD3, the batch size, buffer size and learning rate were tuned. After several semi-manual iterations to try to find the optimal values for each hyperparameter through grid-search, the final round of iterations had the following grid for each model:

- **A2C**
    - Number of steps until update: 5, 10 and 15
    - Learning rate: 0.0001, 0.0002, 0.0003, 0.0004, 0.0005, 0.001, 0.005
- **DDPG and TD3**
    - Batch size: 32, 64, 128 and 256
    - Buffer rate: 10.000 and 100.000
    - Learning rate: 0.0005, 0.001 and 0.005

The main insights are that financial features don't seem to have helped the A2C model in the optimization process. Also, with financial features the DDPG model needed a smaller buffer size (100.000 vs 10.000) and both the DDPG and TD3 found the same solution – and using a simpler model with a smaller buffer requires less computational power, thus being more efficient.

The evaluation stage will be further developed in the "Results and Discussion" chapter. Nevertheless, here the main focus was on the results arising from the backtest and the test set results (especially the Sharpe Ratio – table 6).

	Number of steps until update	Learning rate	Batch size	Buffer size
Market-only features Dataset				
A2C	10	0.001	NA	NA
DDPG	NA	0.001	32	100.000
TD3	NA	0.0005	64	10.000
All features Dataset				
A2C	10	0.001	NA	NA
DDPG	NA	0.001	32	10.000
TD3	NA	0.001	32	10.000

Table 5: Optimized hyperparameters detailed

	SLSQP	A2C	DDPG	TD3
Type	Baseline	DRL	DRL	DRL
Datasets Used	None (closing price)	Market Features + All Features Datasets	Market Features + All Features Datasets	Market Features + All Features Datasets
Hyperparameters Tunned	None	Number of steps until update; learning rate	Batch size; buffer size; learning rate	Batch size; buffer size; learning rate
Evaluation Methods	Returns, Sharpe Ratio	Returns, Sharpe Ratio, Backtest	Returns, Sharpe Ratio, Backtest	Returns, Sharpe Ratio, Backtest

Table 6: Models Descriptive Summary

5. Results and Discussion

This chapter will analyze the results obtained from the built models and discuss them. It starts with the benchmark model, then it goes to the performance of models with market features only, and to the performance of models with the market and fundamental features. As for performance metrics, it was used total performance; sharpe ratio; and backtest performance – all drawn from the test set with 31 months. It's also worth noting that the test set occurred during the COVID-19 pandemic (figure 7), which wasn't around during the training period of the models and can be considered a black-swan type of event. Therefore, this abrupt change in market conditions can provide a good environment to access the models' generalization capabilities.

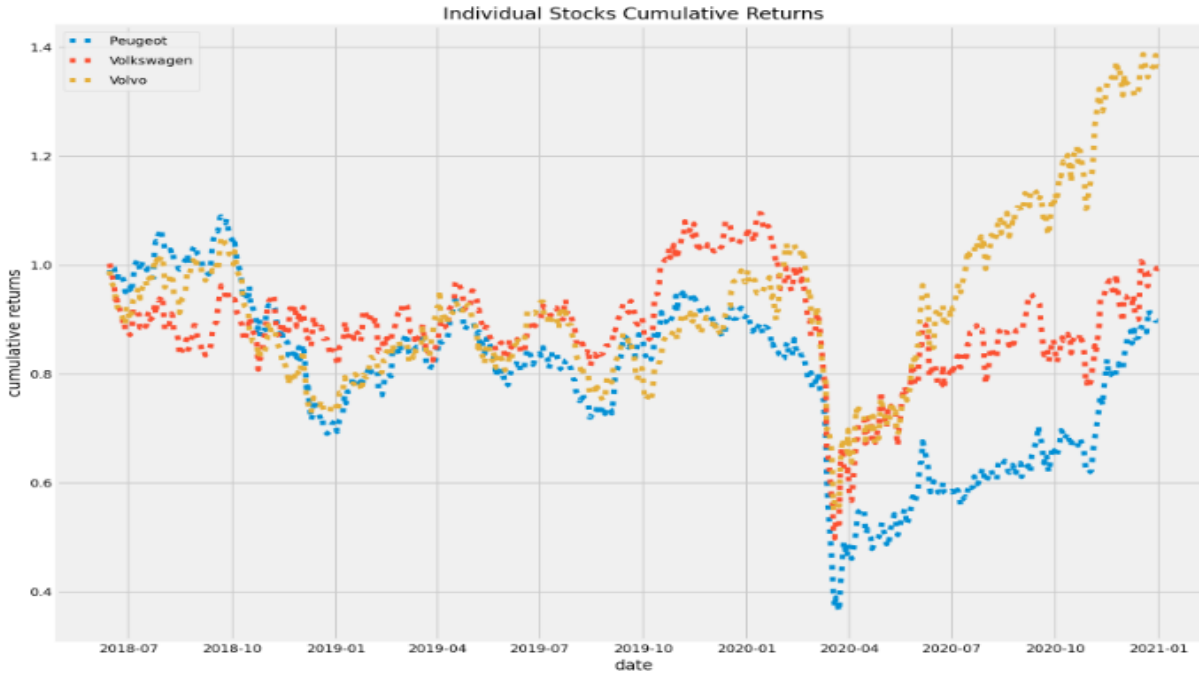

Figure 7: Individual portfolio companies' compounded returns, in the test set period

The backtesting was performed using the *pyfolio* python package in the test set, and these models are then compared with a simple buy-hold strategy of the DJIA (*Dow Jones Industrial Average*) index that comes with the package. The backtest is conducted so that this study can have a higher level of reliability regarding the results achieved. It's an excellent way to understand if the results obtained in the normal test set performance are trustworthy and not a "one time" outlier performance value.

5.1 Baseline Model Performance

Starting with the baseline model, as the name indicates, this model based on traditional optimization techniques is used as the base-case to assess the DRL model's performance. Using the daily returns of each company, the model generated a maximum sharpe ratio portfolio comprised of static weights (mentioned in the "Methodology" chapter). The baseline model portfolio outperforms two out of the three individual stocks comprised in it (figure 8). It has a total return of 12.43% in the test set, while Volvo has 36.48%, Volkswagen has -5.12% and Peugeot has -10.25%.

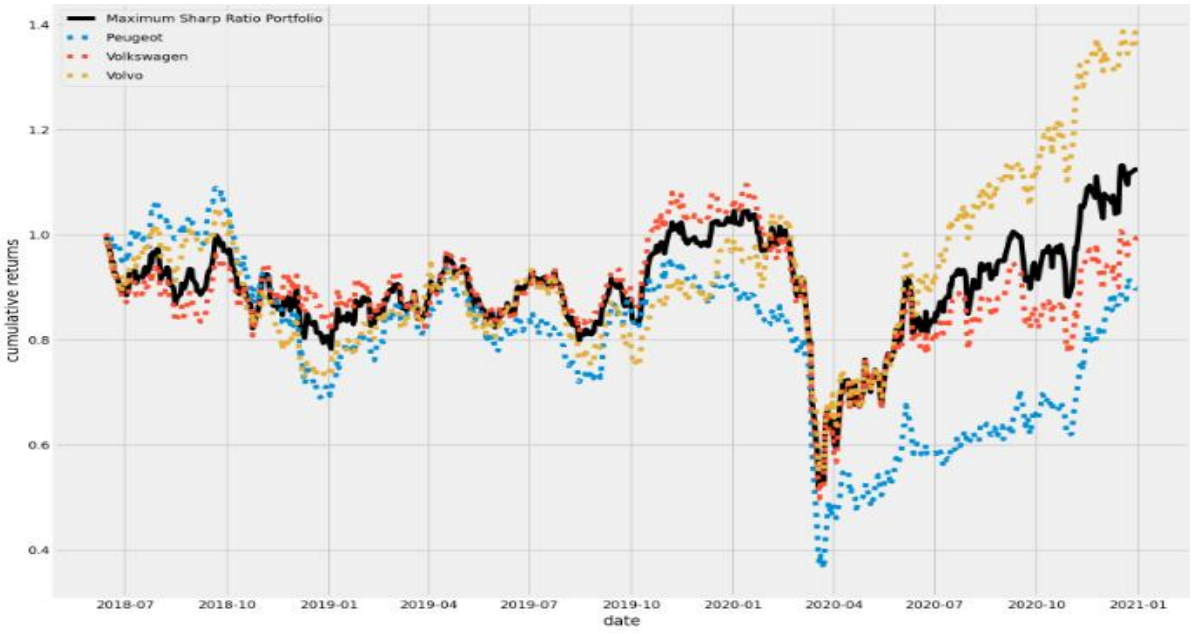

Figure 8: Baseline model portfolio & individual companies' compounded returns, in the test set period

Although Volvo is the best performer in the test set, the portfolio has a higher concentration of Volkswagen shares because in the train set it presented the highest daily mean return of 7.45 x 10^{-4}. Regarding volatility, Volvo and Volkswagen had similar behavior in the train set, as seen in the table below.

	Volkswagen	Volvo	Peugeot
Compounded Return *(test set)*	-5.12%	36.48%	-10.25%
Average Daily Return *(train set)*	7.45 x 10^{-4}	5.99 x 10^{-4}	4.19 x 10^{-4}
Daily Variance *(train set)*	5.94 x 10^{-4}	5.87 x 10^{-4}	4.39 x 10^{-4}

Table 7: Individual portfolio companies performance metrics

5.2 DRL Models Performance

Going to the models' performance on the market-only features dataset (*table 8*), all DRL models had similar performances on the simple test set analysis and backtested results – which helps confirm that these are stable results. No model outperformed the DJIA, nevertheless, all but one outperformed the baseline model (SLSQP). DDPG was the only model that underperformed the baseline, by 73 bps. On average, DRL models had a performance of 15.67% on the test set, which outperforms the baseline model by 324 bps. The average sharpe ratio of all DRL models was also higher than the baseline (avg. 0.35 vs 0.30).

It's important to note that despite the DDPG underperformed against the baseline, both sharpe ratios are the same at 0.30 – meaning that both models returned the same performance (above the risk-free) per unit of risk.

Model	Period Performance	Sharpe Ratio	Backtest Performance
A2C	18.83%	0.38	19.69%
DDPG	11.70%	0.30	11.57%
TD3	16.47%	0.36	16.37%
Baseline	12.43%	0.30	None
DJIA	19.84%	None	None

Table 8: Model performance with market features only

The best model was the Advantage Actor-Critic (A2C), with a test set performance of 18.83% (640 bps higher than baseline) and a 0.38 sharpe ratio (0.08 higher than baseline). The following figures (9 and 10) show the backtest and test set performance, respectively. In figure 9 we can see the logarithmic returns of the backtest against the compounded DJIA daily returns, and the daily returns generated by the backtest on the A2C model. In figure 10 we can see the model's test set performance, against the baseline model and the DJIA index.

The backtest can be considered successful, because the performance achieved is similar to the one on the test set (*table 8*) and the returns' pattern doesn't diverge from the DJIA benchmark comparison – in case divergence happened, it had to be accessed the possibility that the model didn't generalize well enough and that the model "exploded" when it encountered the new environment caused by the COVID-19 pandemic. Another interesting point to analyze, from figure 9, is that daily return's volatility spiked in Q2 2020 (when the pandemic started) and remained higher throughout 2020 compared to the past two years.

Figure 9: A2C Backtest cumulative log-returns (vs DJIA) & daily returns

The A2C model also showed a well-behaved performance pattern in the test set. And despite the total compounded return of the model being smaller than the DJIA performance (18.83% vs 19.84%), the A2C slightly outperformed the index in November 2019 and November 2020.

Figure 10: A2C Model Performance on Test Set, with market only features

Moving on to the performance of models with all features (*table 9*) (market + financial fundamentals). Once again, the similarity of results in the test set analysis and backtest demonstrates consistency/robustness of the deep reinforcement learning models and

trustworthiness of their results. All models outperformed the baseline, but none again outperformed the DJIA index. On average, DRL models had a performance of 15.39% on the test set, which outperforms the baseline model by 296 bps. The average sharpe ratio of all DRL models was also higher than the baseline (avg. 0.35 vs 0.30).

Another key point from this analysis is that both DDPG and TD3 models had the same performance scores – 16.47% period performance, 0.36 sharpe ratio and 16.37% backtest performance. Knowing that TD3 is an enhancement of the DDPG model, it's possible to assume that both models have similar optimized network parameters and that both had a similar (if not the same/identical) understanding of the agent's environment – both taking alike actions under the same environment states.

Model	Period Performance	Sharpe Ratio	Backtest Performance
A2C	13.24%	0.32	13.39%
DDPG	16.47%	0.36	16.37%
TD3	16.47%	0.36	16.37%
Baseline	12.43%	0.30	None
DJIA	19.84%	None	None

Table 9: Model performance with market + fundamental features

Both the Deep Deterministic Policy Gradient (DDPG) and Twin-delayed DDPG (TD3) models had the best performance, with a test set performance of 16.47% (404 bps higher than baseline) and a 0.36 sharpe ratio (0.06 higher than baseline). Figures 11 and 12 show the backtest and test set performance of the DDPG model, respectively. This was the model chosen to be presented in the figures since it is the most original and straightforward model. In figure 11 we can see the logarithmic returns of the DDPG backtest against the compounded DJIA daily returns, and the daily returns generated by the DDPG model backtest. In figure 12 we can see the model's test set performance, against the baseline model and the DJIA index.

Once again, this backtest can be considered successful. Its performance is similar to the one on the test set (*table 9*) and the returns' pattern is similar to the DJIA index – demonstrating that the model generalizes well enough in the new market environment. Also, from looking at figure 11, daily return volatility still spiked in the second quarter of 2020 and remained higher throughout 2020 compared to the past two years – due to the increased market volatility brought by the COVID-19 pandemic.

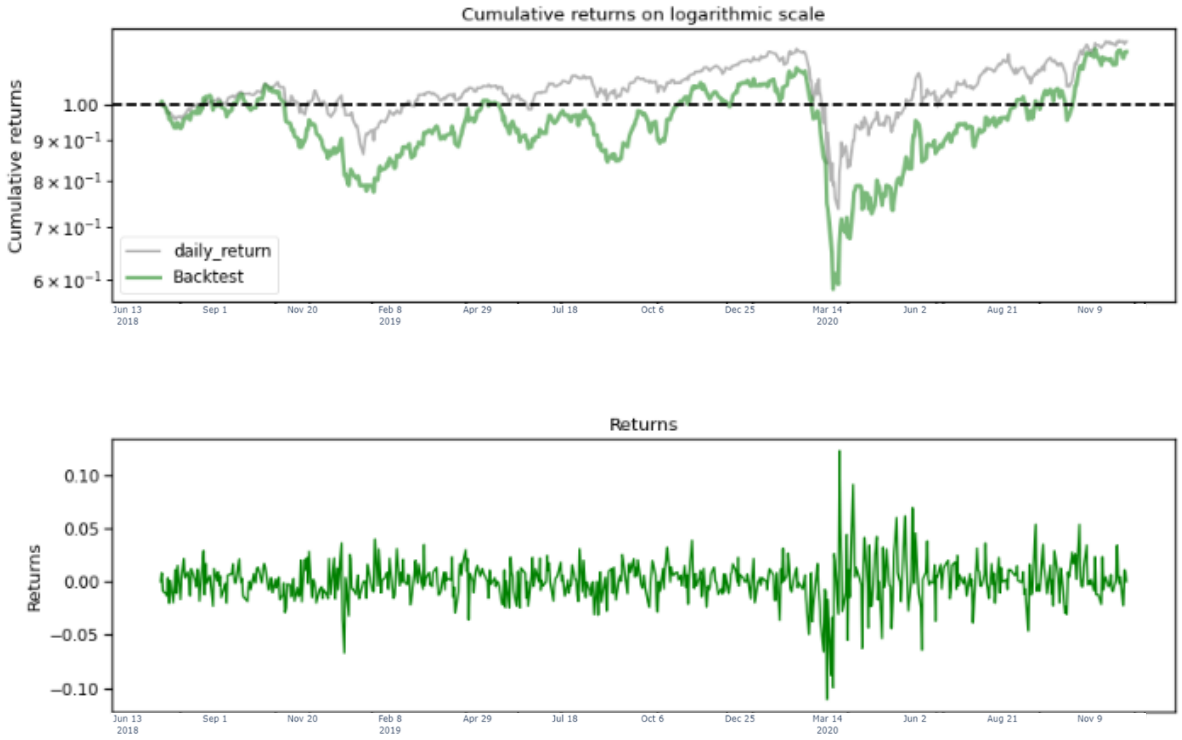

Figure 11: DDPG Backtest cumulative log-returns (vs DJIA) & daily returns

The DDPG model also showed a well-behaved performance pattern in the test set. And despite the total compounded return of the model being smaller than the DJIA performance (16.47% vs 19.84%), the DDPG model was slightly outperforming the index in early November 2020.

Figure 12: DDPG Model Performance on Test Set, with market + fundamental features

5.3 Results Discussion

Overall, deep reinforcement learning models outperformed the baseline model (which consisted of a more traditional quantitative approach). The only model that didn't outperform the baseline was the DDPG with market-only features. Financial fundamental features brought consistency to the models' performance. Although the best model was the A2C with market-only features, this dataset also had the worst performing model. In the market-only features dataset, the spread between best and worst performing DRL model was 7.13% - 18.83% for A2C and 11.70% for DDPG. In the dataset with all features, this spread got reduced to 3.23% - 16.47% for DDPG / TD3 and 13.24% for A2C. Also, despite the average performance of all models in the market-only features dataset having cumulative returns higher than the dataset with fundamental features (avg. 15.67% vs 15.39%), the average sharpe ratio for both datasets remained the same at 0.35. Furthermore, no model with fundamental features underperformed the baseline model.

With these results, we could demonstrate that the portfolio company's fundamental features help bring consistency to the DRL model's performance – enhancing the model's trustworthiness and reliability. Fundamental features also allow achieving the same results in less complex models, which helps to improve transparency in the portfolio allocation process - this is an important aspect for portfolio management and finance generally that is also studied under the emerging field of Explainable AI (XAI) (Dosilovic *et al*., 2018). With fundamental features both DDPG and TD3 had the same performance, whilst with market-only features DDPG underperformed TD3 by roughly 5 percentage points.

This discussion focused on the average performance across all models for both datasets, it was done this way because of two main reasons: i) avoiding focusing the discussion on one single model that might have benefited from the specific environment characteristics provided in this study; ii) one of the two main goals of this study was to understand if fundamental features enhance model's performance and generalization capabilities – not to see if model *a* performed better than model *b*. And in fact, these results show that fundamental features can improve the stability and reliability of DRL models, which generally outperform traditional quantitative methods.

6. Conclusion

As stated in the introduction, this study focused on proving the importance of deep reinforcement learning models in portfolio management and assessing if financial fundamental features enhance model performance. A literature review was conducted where we could see that the scientific community has recently used DRL to solve financial portfolio optimization problems, raising the importance of this study. Here, one could see that multiple DRL models were implemented, but there's a tendency towards using actor-critic DRL models.

Still in the literature review, emphasis was given to the lack of importance feature engineering has on current financial DRL research. This might happen due to the type of problem being solved. Once the AI scientific community finds a new problem to tackle (i.e., portfolio management), there's a tendency to focus on finding and optimizing the best model – which leads to higher model complexity, increased computer power and budgetary needs. If more focus is given to feature engineering while solving new problems, improved data quality can provide an efficient way to obtain good results with simpler models.

After the importance of this study was proven with the literature review, the CRISP-DM methodology was used to conduct the practical experiments. In this stage, three actor-critic DRL models (A2C, DDPG and TD3) were applied and optimized to solve the portfolio allocation problem – using a long-only portfolio with three stocks of companies in the automotive industry. These models were optimized on two different datasets, one with market features (i.e., price, volume traded, technical indicators) and the other with market plus financial fundamental features (i.e., revenue, profit, debt). To conclude, these models were evaluated against a baseline model and were subject to a backtest.

These experiments showed that DRL models outperformed the traditional portfolio optimization technique. The baseline model had a cumulative return of 12.4% in the test set, whilst the DRL models had an average cumulative return of roughly 15.5% - translating to a sharpe ratio of 0.30 for the baseline, against an average of 0.35 for the DRL models. It can be concluded that financial fundamental-based features improve model consistency and robustness, at least when in a portfolio of companies from the same industry, which helps in raising the importance of the emerging field of *quantamental* investments.

This study contributes to explaining the current increasing trend of Portfolio Managers (PM) using Machine Learning (and Deep Reinforcement Learning) models in their portfolio allocation process (López de Prado, 2018, p. 4). Machine Learning provides a better structured and systematic approach to the decision-making process. Quantamental investment strategies

can be a good opportunity for PMs to generate new alpha for their investors and create trust with them through increased transparency. Quoting Marco López de Prado, "*No human is better at chess than a computer. And no computer is better at chess than a human supported by a computer. Discretionary PMs are at a disadvantage when betting against an ML algorithm, but it is possible that the best results are achieved by combining discretionary PMs with ML algorithms. This is what has come to be known as the "quantamental" way*" (López de Prado, 2018, p. 15).

To the scientific community currently trying to solve this problem with DRL, this study serves as another piece of evidence of the benefits that deep reinforcement learning brings to the financial portfolio optimization problem. It may also demonstrate how improving feature engineering can contribute to a better learning environment for the agent to develop its strategy. With simpler models, less computer power required and higher understanding of the problem at hand – in this case, understanding financial assets from different classes and how it relates to portfolio management - one can also contribute to state-of-the-art research. This idea of focusing on data and feature engineering to solve AI problems (instead of model optimization) has been growing, especially with a recent movement, brought by the scientist Andrew Ng, called Data Centric AI (Strickland, 2022).

Regarding the main limitations of this study and the future research agenda recommended. Time constraints and data availability were the two main limitations. Reinforcement Learning is a vast and growing research field. To perform this study, the author had to learn RL from scratch. Meaning that if time hadn't been a constraint, a deeper understanding of RL could have been gained by the author and better/deeper analysis throughout the entire study could have been made – despite the interesting contributions that this study already provides. Data availability was also an important limitation, financial fundamental features had to be taken from Bloomberg terminal, which has associated monetary costs. This is the main reason for the portfolio constructed having only three companies all from the same industry – since the portfolio was small, companies from the same industry were chosen to have a proper assessment of the importance that fundamental features bring. This data availability constraint limits the conclusions that can be taken, especially regarding the importance of fundamental features in a portfolio of companies from diverse industries.

A future research agenda could include building a customized python package with deep reinforcement learning models applied in portfolio management, bringing a higher level of transparency and control over the models by the researcher. A wider range of DRL models can be tried, not just the main actor-critic ones, and assess if the importance of financial

fundamental features persists. Still, on the modelling side, a more extensive hyperparameter tuning can be done to optimize the agents' networks. Feature engineering can always be improved. One could find more interesting markets or fundamental features to enhance model performance. At last, wider portfolios could be tried. These could be cross-industry portfolios, multi-asset portfolios (i.e., with swaps, cash equities and bonds) or even a portfolio with long and short strategies. Thus, allowing future research to better understand the importance of *quantamental* strategies in portfolio management.

7. Bibliography

Aboussalah, A. M., & Lee, C.-G. (2020). Continuous control with Stacked Deep Dynamic Recurrent Reinforcement Learning for portfolio optimization. *Expert Systems with Applications*, *140*, 112891. https://doi.org/10.1016/j.eswa.2019.112891

Abrate, C., Angius, A., de Francisci Morales, G., Cozzini, S., Iadanza, F., Puma, L. L., Pavanelli, S., Perotti, A., Pignataro, S., & Ronchiadin, S. (2021). Continuous-Action Reinforcement Learning for Portfolio Allocation of a Life Insurance Company. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 12978 LNAI* (pp. 237–252). Springer Science and Business Media Deutschland GmbH. https://doi.org/10.1007/978-3-030-86514-6_15

Betancourt, C., & Chen, W.-H. (2021). Deep reinforcement learning for portfolio management of markets with a dynamic number of assets. *Expert Systems with Applications*, *164*, 114002. https://doi.org/10.1016/j.eswa.2020.114002

de Spiegeleer, J., Madan, D. B., Reyners, S., & Schoutens, W. (2018). Machine learning for quantitative finance: fast derivative pricing, hedging and fitting. *Quantitative Finance*, *18*(10), 1635–1643. https://doi.org/10.1080/14697688.2018.1495335

Dixon, M. F., Halperin, I., & Bilokon, P. (2020). Machine learning in finance: From theory to practice. In *Machine Learning in Finance: From Theory to Practice*. Springer International Publishing. https://doi.org/10.1007/978-3-030-41068-1

dos Santos, S. F., & Brandi, H. S. (2017). Selecting portfolios for composite indexes: application of Modern Portfolio Theory to competitiveness. *Clean Technologies and Environmental Policy*, *19*(10), 2443–2453. https://doi.org/10.1007/s10098-017-1441-y

Dosilovic, F. K., Brcic, M., & Hlupic, N. (2018). Explainable artificial intelligence: A survey. *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2018 - Proceedings*, 210–215. https://doi.org/10.23919/MIPRO.2018.8400040

Fu, Z., Liu, G., & Guo, L. (2019). Sequential Quadratic Programming Method for Nonlinear Least Squares Estimation and Its Application. *Mathematical Problems in Engineering*, *2019*. https://doi.org/10.1155/2019/3087949

GAO, N., HE, Y., JIAO, Y., & CHANG, Z. (2021). Online Optimal Investment Portfolio Model Based on Deep Reinforcement Learning. *2021 13th International Conference on Machine Learning and Computing*, 14–20. https://doi.org/10.1145/3457682.3457685

Gao, Z., Gao, Y., Hu, Y., Jiang, Z., & Su, J. (2020). Application of Deep Q-Network in Portfolio Management. *2020 5th IEEE International Conference on Big Data Analytics (ICBDA)*, 268–275. https://doi.org/10.1109/ICBDA49040.2020.9101333

Gu, F., Jiang, Z., & Su, J. (2021). Application of Features and Neural Network to Enhance the Performance of Deep Reinforcement Learning in Portfolio Management. *2021 IEEE 6th International Conference on Big Data Analytics (ICBDA)*, 92–97. https://doi.org/10.1109/ICBDA51983.2021.9403044

Hansen, K. B. (2021). Model Talk: Calculative Cultures in Quantitative Finance. *Science Technology and Human Values*, *46*(3), 600–627. https://doi.org/10.1177/0162243920944225

Harnpadungkij, T., Chaisangmongkon, W., & Phunchongharn, P. (2019). Risk-Sensitive Portfolio Management by using Distributional Reinforcement Learning. *2019 IEEE 10th International Conference on Awareness Science and Technology (ICAST)*, 1–6. https://doi.org/10.1109/ICAwST.2019.8923223

Hu, Y.-J., & Lin, S.-J. (2019). Deep Reinforcement Learning for Optimizing Finance Portfolio Management. *2019 Amity International Conference on Artificial Intelligence (AICAI)*, 14–20. https://doi.org/10.1109/AICAI.2019.8701368

Huang, S. H., Miao, Y. H., & Hsiao, Y. T. (2021). Novel Deep Reinforcement Algorithm with Adaptive Sampling Strategy for Continuous Portfolio Optimization. *IEEE Access*, *9*, 77371–77385. https://doi.org/10.1109/ACCESS.2021.3082186

Jiang, Z., & Liang, J. (2017). Cryptocurrency portfolio management with deep reinforcement learning. *2017 Intelligent Systems Conference (IntelliSys)*, 905–913. https://doi.org/10.1109/IntelliSys.2017.8324237

Kang, Q., Zhou, H., & Kang, Y. (2018). An asynchronous advantage actor-critic reinforcement learning method for stock selection and portfolio management. *ACM International Conference Proceeding Series*, 141–145. https://doi.org/10.1145/3291801.3291831

Khemlichi, F., Chougrad, H., Khamlichi, Y. I., el Boushaki, A., & ben Ali, S. E. (2020). Deep deterministic policy gradient for portfolio management. *Colloquium in Information Science and Technology, CIST*, *2020-June*, 424–429. https://doi.org/10.1109/CiSt49399.2021.9357266

Lee, J., Kim, R., Yi, S.-W., & Kang, J. (2020). MAPS: Multi-Agent reinforcement learning-based Portfolio management System. *29th International Joint Conference on Artificial Intelligence, IJCAI 2020*, 4520–4526.

Leković, M. (2021). Historical development of portfolio theory. *Tehnika*, *76*(2), 220–227. https://doi.org/10.5937/tehnika2102220l

Lin, F., Wang, M., Liu, R., & Hong, Q. (2020). A DDPG Algorithm for Portfolio Management. *Proceedings - 2020 19th Distributed Computing and Applications for Business Engineering and Science, DCABES 2020*, 222–225. https://doi.org/10.1109/DCABES50732.2020.00065

López de Prado, M. (2018). *Advances in Financial Machine Learning*. John Wiley & Sons, Inc.

Lord, M. (2020). University Endowment Committees, Modern Portfolio Theory and Performance. *Journal of Risk and Financial Management*, *13*(9), 198. https://doi.org/10.3390/jrfm13090198

Lucarelli, G., & Borrotti, M. (2020). A deep Q-learning portfolio management framework for the cryptocurrency market. *Neural Computing and Applications*, *32*(23), 17229–17244. https://doi.org/10.1007/s00521-020-05359-8

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., & Kavukcuoglu, K. (2016). *Asynchronous Methods for Deep Reinforcement Learning*. http://arxiv.org/abs/1602.01783

Park, D. Y., & Lee, K. H. (2021). Practical Algorithmic Trading Using State Representation Learning and Imitative Reinforcement Learning. *IEEE Access*, *9*, 152310–152321. https://doi.org/10.1109/ACCESS.2021.3127209

Pham, U., Luu, Q., & Tran, H. (2021). Multi-agent reinforcement learning approach for hedging portfolio problem. *Soft Computing*, *25*(12), 7877–7885. https://doi.org/10.1007/s00500-021-05801-6

Ren, X., Jiang, Z., & Su, J. (2021). The Use of Features to Enhance the Capability of Deep Reinforcement Learning for Investment Portfolio Management. *2021 IEEE 6th International Conference on Big Data Analytics, ICBDA 2021*, 44–50. https://doi.org/10.1109/ICBDA51983.2021.9403019

Shi, S., Li, J., Li, G., & Pan, P. (2019). A multi-scale temporal feature aggregation convolutional neural network for portfolio management. *International Conference on Information and Knowledge Management, Proceedings*, 1613–1622. https://doi.org/10.1145/3357384.3357961

Soleymani, F., & Paquet, E. (2020). Financial portfolio optimization with online deep reinforcement learning and restricted stacked autoencoder—DeepBreath. *Expert Systems with Applications*, *156*. https://doi.org/10.1016/j.eswa.2020.113456

Soleymani, F., & Paquet, E. (2021). Deep graph convolutional reinforcement learning for financial portfolio management – DeepPocket. *Expert Systems with Applications*, *182*. https://doi.org/10.1016/j.eswa.2021.115127

Strickland, E. (2022). *Andrew Ng, AI Minimalist: The Machine-Learning Pioneer Says Small is the New Big*. https://doi.org/10.1109/MSPEC.2022.9754503

Sun, R., Jiang, Z., & Su, J. (2021). A Deep Residual Shrinkage Neural Network-based Deep Reinforcement Learning Strategy in Financial Portfolio Management. *2021 IEEE 6th International Conference on Big Data Analytics, ICBDA 2021*, 76–86. https://doi.org/10.1109/ICBDA51983.2021.9403210

Szepesvári, C. (2009). *Algorithms for Reinforcement Learning*. Morgan & Claypool Publishers.

Tsantekidis, A., Passalis, N., & Tefas, A. (2021). Diversity-driven knowledge distillation for financial trading using Deep Reinforcement Learning. *Neural Networks*, *140*, 193–202. https://doi.org/10.1016/j.neunet.2021.02.026

Venkataramani, R., & Kayal, P. (2021). Systematic investment plans vs market-timed investments. *Macroeconomics and Finance in Emerging Market Economies*. https://doi.org/10.1080/17520843.2021.1969086

Verdiyanto, R. N., Sudrajad, O. Y., & Meyriska, F. (2020). An Empirical Implementation of Markowitz Modern Portfolio Theory on Indonesia Sharia Equity Fund: A Case of Bahana

Icon Syariah Mutual Fund. *Journal of Accounting and Finance in Emerging Economies*, *6*(4). www.publishing.globalcsrc.org/jafee

Warren, G. (2016). *What Does It Mean to Be a Long-Term Investor?* https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2987949

Xu, F., & Tan, S. (2020). Dynamic Portfolio Management Based on Pair Trading and Deep Reinforcement Learning. *ACM International Conference Proceeding Series*, 50–55. https://doi.org/10.1145/3440840.3440861

Zhang, H., Jiang, Z., & Su, J. (2021). A Deep Deterministic Policy Gradient-based Strategy for Stocks Portfolio Management. *2021 IEEE 6th International Conference on Big Data Analytics, ICBDA 2021*, 230–238. https://doi.org/10.1109/ICBDA51983.2021.9403049