A Work Project, presented as part of the requirements for the Award of a Master's degree in Finance from the Nova School of Business and Economics and Universidad de los Andes

PORTFOLIO OPTIMIZATION IN CRYPTOCURRENCIES: A COMPARISON OF DEEP REINFORCEMENT LEARNING AND TRADITIONAL APPROACHES

CÉSAR CAMILO GARCÍA PARDO

Work project carried out under the supervision of:

Melissa Prado

16/12/2022

Abstract

Cryptocurrencies have become appealing investment options in recent years because of their high potential returns. This asset class emerged as a unique investment opportunity with distinguishing characteristics such as decentralized nature and uncorrelation with other assets. Investing in this product, however, has become a hazardous venture due to its extreme volatility and unpredictable price swings. As a result, a portfolio optimization is an essential tool for investors seeking to reduce risk while aiming for high returns. This thesis studies the Deep Reinforcement Learning models applied to cryptocurrency portfolio optimization compared to traditional methodologies like Markowitz's and rudimentary equally weighted portfolios.

Keywords: Cryptocurrency, Decentralized, Deep Reinforcement learning, Markowitz's Optimization, Portfolio Optimization.

1. Outline

The investing environment is ever-changing and has gotten more complicated over time, with new possibilities and risks arising daily. Investors are seeking ways to increase their earnings while minimizing risk. As a result, **portfolio optimization** has become an indispensable tool.

Some traditional portfolio optimization approaches, such as the **Modern portfolio theory (MPT)**, are still frequently utilized by investors. However, advancements in fields such as artificial intelligence (AI) have derived new portfolio optimization technologies. For example, **Deep Reinforcement Learning (DRL)** has resulted in innovative portfolio optimization strategies.

All terms underlined in bold are critical to the project and will be discussed further in this text. For the time being, an explanation of the thesis's purpose is to be given, followed by the hypothesis and a question that may emerge when considering the issue under study.

This thesis analyses and contrasts portfolio optimization approaches in cryptocurrencies utilizing DRL versus MPT. A review of the benefits and drawbacks of each technique has also been included, allowing the reader to choose which strategy is ideal for different types of investors depending on their preferred degree of risk or expected returns.

A distinction between strategies relies on the fact that conventional approaches depend on past data to show correlations between assets and estimate future returns (Jin, Li, and Yuan 2021). As a result, these techniques that rely on past data frequently based on statistical models may be inaccurate in specific instances, such as the volatile cryptocurrency market. Meanwhile, the DRL methodology relies on experience rather than historical data, allowing DRL to adapt to changing market conditions and discover better answers than traditional methodologies (Sutton and Barto 2017).

The hypothesis states that portfolio optimization based on DRL will outperform traditional methodologies, being more efficient and effective in risk management than, for example,

Markowitz's approach. The premise is that DRL algorithms may learn from experience and adapt to changing market conditions, thus being more effective than classical algorithms.

Hence, a question that may arise as a result of this method of portfolio optimization is:

1. What is the optimal currency portfolio for a crypto-investor?

To give some insights to the reader, the obtained findings for this working project point in the same direction as the hypothesis, except for one instance in which one of the traditional techniques outperformed one of the DRL algorithms; section four of the working project will explain this exception and all of the obtained results.

For the time being, the next section will give a literature review to support the hypothesis developed for this working project (DRL outperforming traditional methodologies). After, because the thesis focuses mainly on cryptocurrencies, a brief description of their features is required. It is important to remember that some key terms, such as portfolio optimization, DRL, and the Markowitz approach, will be used in this section and will be discussed in a simplified manner, but a separate section for those terms will be provided later on the work due to their importance for the work's outcomes.

1.1 Literature review

The optimization method suggested by Harry Markowitz in 1952 was the only technique available in the early days for portfolio optimization. This approach is still widely employed but has significant flaws, such as presuming normally distributed asset returns[1], which is not the case in the actual world; also, it assumes all investors are risk-averse[2], which implies they are inclined to trade off profits to reduce risk (Karandikar 2019). However, many investors nowadays are risk seekers, meaning they are prepared to take on higher threats to make more

[1] Normally distributed asset return follows a bell curve, which means that the vast majority of returns revolves around the mean, with only a tiny number of yields being either greater or less than the mean.

[2] Risk aversion is a concept in economics and finance that expresses a dislike for risk based on human behavior (particularly that of consumers and investors). People have a propensity to favor certainty over ambiguity. As a result, someone who is risk-averse is hesitant to take chances.

money.

In recent years, there has been an increase in interest in applying DRL to optimize portfolios. The benefit of using DRL for optimization problems is that it does not rely on robust assumptions, unlike the Markowitz approach, and it can handle non-linear issues[3], which are common in portfolio optimization problems. Furthermore, it is suitable for dealing with numerous objectives at the time, including returns maximization and risk mitigation (Benhamou et al. 2020) (Gemechu 2020).

Several studies in the literature compare traditional optimization methodologies to DRL. The results of these studies reveal what was mentioned previously in the hypothesis, especially when traditional methods' assumptions are not satisfied.

First, (Benhamou et al. 2020) investigated the portfolio allocation problem using DRL. The authors emphasized that DRL can have a better adaption process to today's changing market conditions, and over time, the DRL approach can include new knowledge to make future judgments. As a result, AI-based models will improve.

Sadriu (2022) did similar research, optimizing a stock portfolio based on the Swedish Stock Exchange (OMXS30) using DRL. Two well-known algorithms were employed in this research: Advantage Actor Critic (A2C) and Deep Determinist Policy Gradient (DDPG), they will be explained in detail later in this work. The author concludes that both algorithms outperformed traditional techniques in two crucial areas: risk and returns throughout the five years covered by the research.

Similarly, (Yang et al., n.d.) did another study, again using the previously mentioned algorithms, but this time the Dow Jones 30 (DJI) was used as the trading stock pool. One can

[3] A non-linear problem is one in which the intended outcome cannot be determined simply by a linear combination of inputs. These might suggest that the issue is non-convex, in this case using standard optimization techniques does not lead to finding a single global optimum, and thus, iterative methods are needed to find the optimal solutions.

see in the investigation conclusions similar results to the previously stated works in which DRL seems to outperform traditional approaches.

As shown, some studies support using DRL approaches for portfolio optimization. One of the motivations for this thesis is to assess the influence of DRL algorithms in a volatile environment such as the crypto world.

1.2 Cryptocurrencies

In recent years, cryptocurrency has surged in popularity. Bitcoin and Ethereum, for example, are digital assets that employ cryptography[4] to protect user transactions and limit the creation of new asset units. Cryptocurrencies are often traded with decentralized transactions[5] and may be used to purchase products and services (Houben, and Snyers, 2018).

Because of its decentralized structure, cryptocurrency is supposed to be safer and less vulnerable to financial fraud than traditional currencies. There are several significant distinctions between these financial assets and other fiat currencies[6], such as the first being more volatile than fiat currencies, cryptocurrencies not being backed by governments, and the possibility of using cryptocurrencies anonymously, while fiat currencies cannot (Dapp, Helbing, and Klauser 2021).

One may also explore some of the reasons why cryptocurrencies are so volatile. Figure 1 depicts the volatility of the daily log returns of Bitcoin and Ethereum compared to the S&P500 over the past few years. But fundamentally, some reasons could be that cryptocurrencies are still very new, and as a result, this market sector is still quite volatile. Furthermore, there is speculation about cryptocurrency trading, which may lead to significant price fluctuations.

[4] The practice of secure communication in the existence of other sides is known as cryptography. It may be used for email, file sharing, encrypted messaging, and cryptocurrency trading, among other things.

[5] A decentralized transaction is one in which a central authority does not regulate it. This sort of transaction is frequently peer-to-peer and is not subject to the same laws and regulations as a centralized transaction.

[6] A fiat currency is one that has been declared legal tender by a government but lacks the backing of a tangible commodity such as gold or silver.

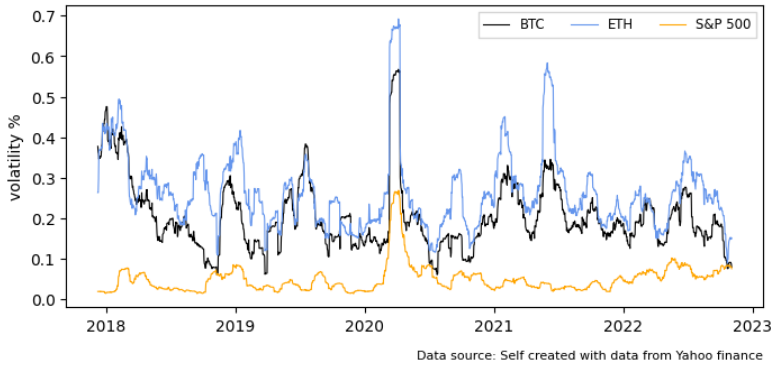


Figure 1. Volatility of daily log returns for BTC, ETH, and S&P 500

Additionally, the uncontrolled environment in which this crypto asset evolves may explain part of its volatility. Finally, the supply of most currently recognized cryptocurrencies is restricted, contributing to price volatility (Doumenis et al. 2021).

After assessing the definitions of the mentioned terms in bold, the document will take the following structure: the methodology will be explained, providing a deep detail of the form in which DRL and any other strategy is to be used to obtain the expected results; followed, the database(s) will be detailed and explained as well as the variables it contains, and the final section will provide the reader with the insights obtained after running the model(s) and the conclusion.

1.3 Portfolio Optimization

"Diversification is an established tenet of conservative investment"
Benjamin Graham

The goal of portfolio optimization is to find the asset mix to produce a portfolio(s) that maximizes returns for a risk level or reduces the risk for a previously defined yield level. Thus, portfolio optimization becomes critical for investors looking to accomplish optimal financial goals that align with their expected returns while considering the risk they are willing to incur. Many authors have examined the issue under discussion in the literature, for example, the Nobel laureate Harry M. Markowitz, who conducted research on portfolio optimization in 1952 by first developing the Modern Portfolio Theory (MPT). The study primarily transformed how people and institutions invest. Markowitz demonstrated that diversification across asset classes might reduce risk without sacrificing return. Other writers validated these findings, becoming one of the core principles of MPT (Ceria and Krishnan Sivaramakrishnan 2013).

Several factors must be addressed when optimizing portfolios, including the investor's risk tolerance, investment objectives, and time horizon. Furthermore, portfolio optimization considers asset correlation, the individual expected return, and the volatility of each asset.

Portfolio optimization benefits include the ability to boost risk-adjusted returns, portfolio diversification, and aiding in the mapping of an investment plan. On the other side, there is a danger of over-diversification. Thus, continual portfolio monitoring and rebalancing are required. Those last can be time-consuming and, if not done correctly, can result in a portfolio that does not meet the investor's objectives (Cy and Polyviou 2020).

1.4 Markowitz portfolio approach

"A good portfolio is more than a long list of good stocks and bonds. It is a balanced whole, providing the investor with protections and opportunities with respect to a wide range of contingencies"
Harry M. Markowitz

When discussing Markowitz's portfolio concept, it is essential to note that it is a mathematical framework that aims to design the optimal portfolio. The model implies that investors are rational and risk-averse (Mangram and Mangram 2013). As previously stated, the aim is to determine the appropriate asset allocations that allow maximum returns while reducing risk. The theory demonstrates a method for quantifying this trade-off and establishing the best portfolio.

The model's notion is that portfolio risk is determined not only by each component's risk sum but instead that the entire portfolio should be evaluated as a whole. The fact that each element is unrelated might explain it. Hence, diversifying each can reduce the overall risk of the portfolio (Mangram and Mangram 2013).

Consider some key identified assumptions underlying Markowitz's portfolio theory: first, investors have free and complete information, implying they are aware of the true nature of returns and risks. Second, investors are rational and choose to optimize a utility function given a specific income. Finally, Investors are risk averse. As a result, they strive to minimize risk while enhancing benefits (Omisore 2012).

The Markowitz portfolio approach relies on the robust premise that two variables describe all assets' behavior: expected returns and risk. The expected returns are the average returns that investors may expect to receive given specific invested money over an explicit timeframe window. Risk represented by the variance is the possibility that an investment can lose value over a certain period. The following calculations employ the (Bodie, Kane, and Marcus 2014) Investment book notations. Equations one and two show how the preset variables appear mathematically for a given portfolio p of n assets:

Variance as:

$$\sigma_p^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} \omega_i \omega_j Cov(r_i r_j) \quad (1)$$

Expected returns as:

$$\mathbb{E}(r_p) = \sum_{i=1}^{n} \omega_i \mathbb{E}(r_i) \quad (2)$$

The square root of the variance is known as the standard deviation:

$$\sigma_p = \sqrt{\sigma_p^2} \quad (3)$$

In all the preceding equations, ω represents the individual weights for each asset i, and r_p represents the portfolio return; returns for individual securities are shown in equation four, while equation five shows portfolio returns.

Individual securities returns:

$$r_{t+1} = \frac{P_{t+1} - P_t}{P_t} \quad (4)$$

and portfolio returns:

$$R_t = \sum_{i=1}^{n} \omega_{ti} \times r_{ti} \quad (5)$$

P signifies the price, and t specifies the time for individual financial assets (the reader might also find the optimization problem equations in matrix notation in Appendix A). The main

equations required for setting the Markowitz portfolio approach are equations one through five; the optimization problems for each of the selected traditional methods are displayed later in the methodology; for the time being, this was an introduction to the Markowitz approach to encourage the reader with the basics of the model, and now some insights about the DRL methods are to be given.

1.5 Deep Reinforcement learning (DRL)

"A breakthrough in machine learning will be worth ten Microsoft's"
Bill Gates

Deep Reinforcement Learning is a subfield of ML. Agents in DRL algorithms learn by acting in an environment and obtaining rewards or punishment for their actions. This process may be understood in the same manner humans learn: via trial and error.

DRL algorithms enable agents to identify how to attain various objectives using the original data (inputs) by employing a neural network, a kind of AI inspired by how the human brain functions. To better predict the output of a given information, neural networks learn by adjusting internal parameters known as weights. The neural network weights are modified in DRL to maximize the expected reward of the agent (Cotta et al. 2002).

Until now, it is possible to comprehend the relevance of portfolio optimization. After introducing DRL, one may explore how to use this approach to generate a feasible solution to the problem. As previously stated, the objective is to create a portfolio that optimizes expected profit while limiting risk. To achieve this trade-off, the agent must learn to trade between these two objectives, and the feedback obtained after each trade will aid in this endeavor.

There are several approaches to expressing the Portfolio Optimization Problem (POP) as a DRL problem: In its most basic form, consider an agent that learns which activities will result in the best returns. One example could be an agent trained using historical data from previously generated portfolios to understand which behaviors resulted in the highest returns. As a result, the agent may be used to make various judgments about which stocks from a given list should

be bought or sold to maximize the portfolio's total returns.

The literature provides advanced solutions to this problem. But before we get into them, it's critical to comprehend a few basic terms from the DRL world. One can use the (Arulkumaran et al. 2017) study and (Achiam 2020) OpenAi Python package user documentation to explain these phrases. To begin, the term **agent** refers to software that interacts with an environment to learn how to maximize some concept of cumulative reward. A process of self-improvement increases an agent's ability to select a set of actions that maximizes an expected return[7]. Other terms to consider are i) the **environment** is what the agent interacts with to get rewards or punishments[8]; ii) the collection of all actions an agent can do while interacting with the environment refers to the **action** (a_t) an agent can take. In most typical applications of DRL, the agent may perform discrete actions like moving up, down, to the left, or the right. Furthermore, the agent can do a continuous sequence of operations, such as selecting things from a predefined list of objects[9]; iii) the **state** (S_t) of each feasible environment at time t, which is all the information about the environment that an agent may see. Until this point, the concept will be an agent interacting with its environment by performing actions (a_t) in the state (S_t). Following the conclusion of an action, the environment and the agent enter a new state (S_{t+1}) dictated mainly by the agent's activity and the existing state[10]; iv) Finally, the **policy** is a collection of predetermined rules that the agent uses to select what actions to take in a particular state[11]. When the environment changes state, it also gives the agent a reward of r_{t+1}

[7] Bond trading agents, currency trading agents, and stock trading agents are some examples of DRL agents applied to finance.

[8] Algorithmic trading, portfolio management, and stock trading are a few examples of RL environments applied to finance.

[9] Analyzing financial data to find trends and patterns, identifying and classifying financial data, making forecasts of future economic circumstances, monitoring markets, and recommending potential investment plans are some more tasks an agent in RL can conduct. Furthermore, deciding whether to buy, sell, or hold a stake in an asset.

[10] Some but not all of the RL states used in finance are the current price of a security, the current portfolio value, the current position in a stake, and the time since the last trade.

[11] Market making, portfolio optimization, and risk management are some examples of RL policies in finance.

as evaluation. The agent attempts to get a policy. Or, in other words, a strategy that maximizes expected return. Mathematically the policy can be seen as $a_t = \pi(\cdot|s_t)$.

Figure 2 shows a loop representing the agent-environment interaction. It is a feedback loop that teaches an agent how to respond in a given scenario in order to attain a goal. The loop begins with the agent seeing and reacting to his environment. The agent collects state information from its surroundings and then applies its policy to choose the best action. After, a transitional face starts in which the environment changes a step, providing a next state s_{t+1}. The environment then assesses the action and provides feedback to the agent, frequently in the form of a reward r_{t+1} or a penalty. The agent then uses this information to adjust its behavior and improve its policy.

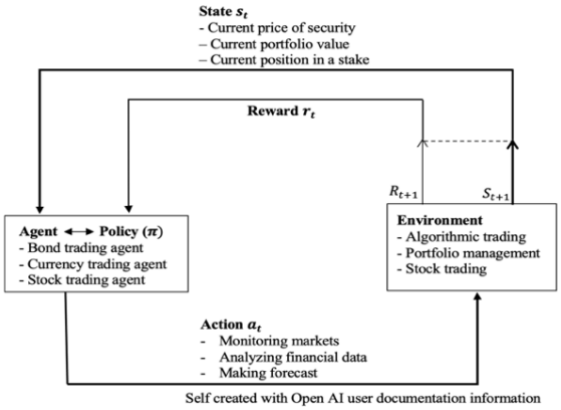


Figure 2. The feedback loop of agents

Under the OpenAi python package user manual and using the same notations, one may also discuss the reward function, mathematically represented as $r_t = (S_t, a_t, s_{t+1} \dots, s_{t+n})$, and the trajectory function, defined by $\tau = (s_0 a_0, s_1 a_1 \dots, s_n a_n)$, which are significant concepts to consider before digging into the DRL optimization problem equations. These last equations demonstrate that both are affected by a state action pair.

With this information, plus the OpenAi python package, equation six is computed, presenting the return function, which is the total of all earned rewards for a certain period.

$$R(\tau) = \sum_{t=0}^{\tau} r_t \qquad 6$$

The DRL problem intends to reach the optimal policy represented by π^*. To visualize the problem, consider the probability distribution for a T - step trajectory. Equation seven illustrates this scenario:

$$P(\tau|\pi) = \rho_0(s_0)\prod_{t=0}^{T-1} P(S_{t+1}|S_t, a_t)\pi(a_t|s_t) \quad 7$$

The expected returns are therefore denoted by $\mathcal{J}(\pi)$:

$$\mathcal{J}(\pi) = \int_{\tau}^{\cdot} P(\tau|\pi)R(\tau) = \underset{\tau\sim\pi}{\mathbb{E}}[R(\tau)] \quad 8$$

Thus, the optimal policy can be calculated from the expected returns equation as follows:

$$\pi^* = arg\max_{\pi} \mathcal{J}(\pi) \quad 9$$

Using the previously described studies, which assisted in explaining the core principles of the DRL universe is possible to extract the information needed to discuss the three DRL methods identified in the literature: **Value-based** algorithms intend to determine the value of a state or action. In other words, its goal is to forecast an agent's anticipated return in a particular condition (how effective it is for the agent to carry out a specific action in an assumed state); **policy-based** algorithms intend to find the best policy π^* for an agent. The algorithm works by searching through a set of alternative policies and then picking the approach that results in the maximum reward for the agent, and **model-based** algorithms in DRL are prediction methods that learn an environment model. Thus, this algorithm employs a model that forecasts the future condition of the environment and then combines all of the gathered data to select the optimum action (Arulkumaran et al. 2017).

1.5.1 DRL groundwork

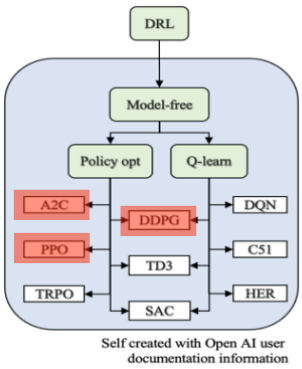


Figure 3. DRL groundwork

Figure 3 offers the groundwork for the reader to understand what follows; moreover, the figure lets the reader see the selected algorithms used in this thesis, but the next section will provide a more detailed explanation. Highlighted in red in figure 3 are the chosen algorithms for this working project which are: Advantage Actor Critic (A2C), Proximal Policy Optimization (PPO), and Deep Deterministic Policy Gradient (DDPG), all of them concerned

with policy optimization and Q-learning approaches.

1.5.1.1 Model-free DRL

The Model-free method in DRL does not require a model of the environment. Therefore, algorithms that use it may learn directly from experience without having to consider creating a model of the environment. This approach is simpler to implement and has a greater learning rate than Model-based methods in which the agent has access to an environment model. Even if the first scenario is simpler to construct, it may take more experience to get the exact extent of efficiency as Model-based solutions (Szepesvári et al. 2009).

Because the Model-free technique does not have access to an environment model, the question of what the agent should learn arose. There are two basic techniques in model-free DRL: policy optimization and Q-learning:

1.5.1.2 Policy Optimization

Policy optimization is a training strategy for DRL agents that focuses on improving the agent's policy. The objective is to determine a policy that permits the agent to receive the highest reward. An iterative process of testing numerous policies before picking the one that provides the best payment for the agent begins. Mathematically the policy is $\pi_\theta = (a|s)$, and the optimization process consists of optimizing the parameter θ and immediately running a gradient ascent on equation eight, as illustrated in the following equivalence $\mathcal{J}(\pi_\theta)$. This optimization is **on-policy** based, meaning it only uses information from the environment consistent with the most recent version of the policy (Degris, Pilarski, and Sutton 2012).

1.5.1.3 Q-learning

Q-learning is a DRL algorithm to find the best policy in a given environment. This technique iteratively updates a Q-table[12] that reflects the predicted reward for each environment's state-

[12] The Q-table is a matrix that contains all possible states and action combinations. Plus, the associated Q-values for each state-activity mix. The Q-values are updated as the agent explores the environment and determines the best policy.

action (appendix B table 1 offers an example of how the Q-table looks). The method tends to converge to an optimal policy when the Q-table iterates up to the correct value function of the environment. Q-learning techniques develop a mathematical approximator $Q_\theta(s|a)$ that leads to the efficient action-value combination $Q_\theta^*(s|a)$. This optimization is **off-policy** based, meaning it can use information from the environment consistent with any version of the policy during the training period. Because this technique employs an additional argument represented by the parameter θ. Q^*, and π^* now decide the optimum approach (Li, Ni, and Chang, n.d.).

2. Methodology

The idea stated as part of the methodology for this working project was to emulate (Durall 2022) study "Asset Allocation: From Markowitz to Deep Reinforcement Learning," in which the author presented traditional approaches for portfolio optimization, discussing different strategies using the Markowitz model as a starting point, and then describes asset allocation paradigm while thinking in DRL and comparing it to traditional approaches.

The difference between this study and the previous one is that the former focuses on cryptocurrencies while the latter does not. Furthermore, because crypto assets are more volatile than traditional assets, this study takes place in a highly volatile setting, allowing us to see the impact of using ML technologies in such a highly volatile setting.

The next section of the methodology goes deeper into how the selected algorithms work, beginning with traditional portfolio optimization methods (Markowitz, Minimum Volatility, Tangency, and Equally Weighted portfolios) and then moving to the three chosen DRL algorithms (A2C, PPO, and DDPG Models).

Before delving more into classic optimization approaches, it is crucial to note a well-known performance indicator identified as the **Sharpe ratio**, which merely refers to a risk-adjusted measure that compares the predicted return of an investment portfolio to its volatility (Schmid and Schmidt 2010). The goal is to get a higher Sharpe ratio, which translates into investors

earning a better return per unit of risk; the following equation shows how to compute the ratio:

$$S = \frac{\mathbb{E}(r_p) - r_f}{\sigma_p} \quad 10$$

Where S is the Sharpe ratio, and r_f is the risk-free rate.

For the sake of this study, the risk-free rate is set equal to the three-month US Treasury bill (4.06%) on November 2022 (FRED 2022). Based on history, the US Treasury bill is considered a highly secure investment due to its support by the US government's faith and credit.[13] The other parameters are the same as in the previous sections.

2.1 Traditional portfolio optimization models

This component of the working project builds on the material offered in part 1.4 by giving further details on the previously disclosed assumptions, parameters, and equations that let the Markowitz portfolio strategy work. To initiate the Efficient Frontier is discussed, which will provide the reader with the knowledge needed to begin the discussion over classical optimization tactics.

2.1.1 Efficient frontier (EF)

The EF is a collection of all optimum portfolios with the best returns for a given amount of risk. All EF-based portfolios are efficient because they deliver the best-expected returns given a risk level. Portfolios below the EF are known to be sub-optimal because they provide lower yields for a certain level of risk; similarly, portfolios over the EF are still sub-optimal because they offer more hazard than is required to get a given level of return. The EF is significant because it may assist investors in determining the best portfolio for their requirements. Furthermore, it can be used to compare various investing approaches (Badea 2008).

The black dotted line in Figure 4 depicts the EF line; the green cross represents the portfolio with the lowest global volatility associated with the least amount of risk; the red cross shows

[13] Furthermore, the three-month Treasury bill in the United States is a highly liquid investment, which means selling or exchanging it can swiftly convert it into cash.

the tangent portfolio, which also illustrates the portfolio with the maximum Sharpe ratio.

Milthaler (2020) Phyton **FinQuant** package, which contains a Monte Carlo[14] simulation function, was used to create this graph. Monte Carlo simulations are used in the portfolio optimization process to determine the best asset allocation in a given portfolio. In this scenario, the simulations were performed 1.5 million times, with each run yielding a distinct set of outcomes, each represented by the colorful dots under the EF line. Each color reflects the Sharpe ratio; the dots closer to red represent the crypto asset allocation with the lowest Sharpe ratio, while the dots near a blue color represent the crypto asset allocation with the highest Sharpe ratio. The green, red, and black triangles reflect the run Monte Carlo simulations' Minimum Volatility, Max Sharpe, and equally weighted portfolio allocations.

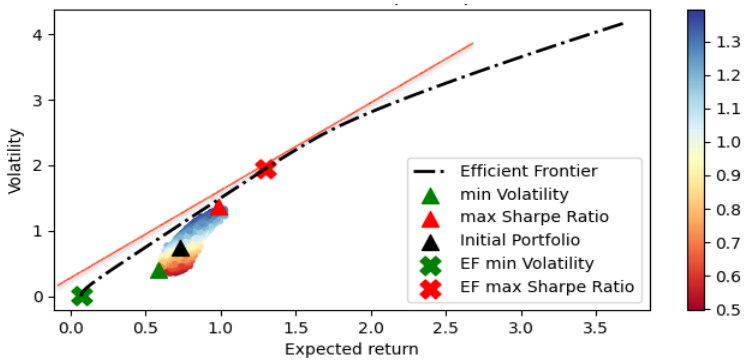


Figure 4. Efficient Frontier of the optimal portfolios and Monte Carlo simulations

The optimal weights for the classical techniques were determined using the Phyton **PyPortfolioOpt** package. This tool requires two main inputs: i) a data frame containing the adjusted close price of all cryptocurrencies needed to calculate the expected returns and covariance matrix of returns for each asset, and ii) the risk-free rate.

2.1.2 Markowitz Portfolio

The Markowitz portfolio relies on the mean-variance approach. A set of assets is selected to lower risk while achieving a target amount of expected return (Bodie, Kane, and Marcus 2014); for this thesis, the target returns introduced on the code were 30%. This method enables investors to optimize their portfolios based on their desired returns. Alternatively, one can

[14] Monte Carlo simulation is a mathematical approach that generates random samplings from a random simulation to provide various probable outcomes for specified conditions. The ultimate objective is to utilize the findings of the random simulations to determine the likelihood of scenarios occurring.

approach the problem in the reverse direction by computing an acceptable level of volatility and seeking to optimize expected returns. The following equation depicts the Markowitz portfolio optimization problem for the first alternative:

$$\min_{\omega} = \frac{1}{2}\sigma_p^2 \ \ \mathbf{s.a} \ \ \omega_r^{\mathrm{T}} = \ \mathbb{E}(r_p) = 30\% \ \ \boldsymbol{and} \ \ \omega^T 1 = 1 \qquad 11$$

2.1.3 Minimum Volatility Portfolio

The Minimum Volatility portfolio is a collection of assets associated with a low level of risk. This portfolio is also known as the Risk-free rate since its primary goal is to minimize risk (Bodie, Kane, and Marcus 2014). The benefit of the least volatility portfolio is that it is less likely to offer short-term losses. However, in most cases, this portfolio will underperform the market in the long run. The following equation depicts the Minimum Volatility portfolio optimization problem:

$$\min_{\omega} = \frac{1}{2}\sigma_p^2 \ \ \mathbf{s.a} \ \ \omega_r^{\mathrm{T}} = \ \mathbb{E}(r_p) \ \ \boldsymbol{and} \ \ \omega^T 1 = 1 \qquad 12$$

2.1.4 Tangency Portfolio

Tangent portfolios are those that are tangent to the efficient frontier; it is also known as the maximum Sharpe portfolio since it provides a combination of assets within a portfolio that maximizes the Sharpe ratio. This EF portfolio has the best risk-adjusted return, which implies that it delivers the maximum potential return per unit of risk (Bodie, Kane, and Marcus 2014). One downside of this strategy is that it is sometimes impossible to execute in practice since it demands precise knowledge of future market circumstances. The following equation depicts the Tangency portfolio optimization problem:

$$\max_{\omega} = \frac{\omega^T r - r_f}{\sigma_p^2} \ \ \boldsymbol{s.a} \ \ \omega^T 1 = 1 \qquad 13$$

2.1.5 Equally weighted portfolio

The Equally Weighted portfolio gives the same weight to each asset regardless of its contribution to the portfolio's total risk or return. The advantage of this portfolio development

strategy is its ease of implementation. The disadvantage compared to the Markowitz portfolio, for example, is that it does not consider the individual risk and return characteristics of the assets in the portfolio, which may result in poor results.

The ideal weights for each method were derived using the previously described Python package based on **CVXPY**, a modeling language used in Phyton for solving convex optimization problems. Part four of the work will present the outcomes. This tool considers all the methods and how they should be weighted to achieve the best and desired results.

For the time being, it is possible to see how each approach has some drawbacks, which stem primarily from the premises that rely on the application of Markowitz's theory, such as rational investors, the use of historical data to generate the optimal portfolio, and the difficulty of correctly quantifying asset volatility.

2.2 Deep Reinforcement learning algorithms

This section of the working project extends the material presented in part 1.5 by offering further details on the previously provided DRL information, but now it is meant to lead to a debate on the three algorithms selected earlier. The findings for the IA-based portfolio optimization strategy were obtained using the Python package (FinRl 2022). **FinRl** is an open-source solution that uses DRL capabilities to solve financial concerns.

The algorithms require the following inputs: i) a data frame including financial information for all selected cryptocurrencies (adj closing price, high, low, and volume). The reader should be aware that, unlike the traditional approaches, this additional information is to be used to add new financial indicators to the data frame necessary for training the DRL algorithms; appendix B table 2 lists the ones employed in this study. ii) a trading environment derived from the previously described Python package, and iii) an agent derived from another Python package (StableBaselines3 2022)**.**

The most significant changes to the required inputs were, for example, adapting the

environment to consider an all-year trading period because the crypto market never closes and including more volatility financial measures to feed the models, all to attempt to attack the highly volatile environment in which crypto assets evolve. One can consider a bunch of financial metrics to nourish the models, but in this case, and because of computational power, it was no longer feasible.

2.2.1 Advantage actor critic (A2C)

Actor Critics is a DRL algorithm that combines value-based and policy-based methodologies. This approach estimates the values for each state and action using a state-value function and an action-value function. The algorithm then utilizes the predicted values to determine the optimal action to take in each state. A2C maximizes its performance by using gradient ascent. Gradient ascent is an optimization approach used to identify a local maximum function; consequently, the process begins with an initial guess of the parameters and improves the guess by iteration until it converges to the ideal value (Mnih et al. 2016).

2.2.2 Proximal Policy Optimization (PPO)

PPO is a DRL algorithm used to optimize a policy. Making minor and gradual modifications to the present policy yields the ideal approach. The phrase "proximal" implies that the algorithm only updates the policy portions closer to the current policy. In other words, the components of the policy that are most likely to affect the policy's present performance. An objective function is optimized, yielding a cautious prediction of how much $\mathcal{J}(\pi_\theta)$ will change due to policy modifications (Schulman et al. 2017).

2.2.3 Deep Deterministic Policy Gradient (DDPG)

Deterministic Policy Gradient is a DRL method used to train agents in continuous spaces. DDPG is an off-policy technique that estimates a Q-function using a Deep Q-network (DQN). For training this model, the Bellman equation[15] is used, which helps to minimize the difference

[15] The Bellman equation is a mathematical framework that leads to the best policy for any Markov decision process (MDP). An MDP is a delimited collection of states, a limited number of actions, and a function that indicates the

between the DQN's predicted and actual Q-values. This technique also utilizes the actor-critic structure, in which the actor training is created by mapping the states to actions using the policy gradient, as in the A2C model, and the critic maps the States of the Q-values, as previously mentioned, by using the Bellman equation (Lillicrap et al. 2015).

3 Data base(s) exploration

Data was downloaded from Yahoo Finance from November 9, 2017, to November 4, 2022. The sample for both the traditional and AI-based techniques includes eighteen of the oldest cryptocurrencies, allowing for more information to be retrieved. Figure 5 illustrates the name of each crypto asset plus the number of data points for each cryptocurrency, which was 1822 for the specified period. In addition, for each cryptocurrency, table 3 in the appendix B presents descriptive statistics.

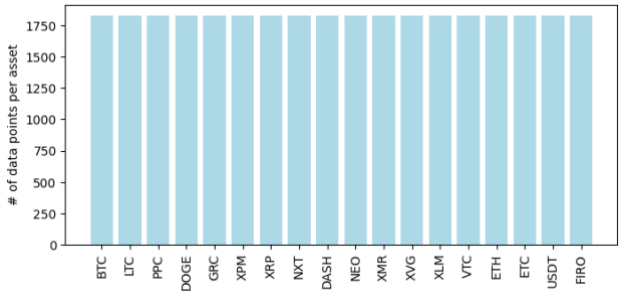


Figure 5 The number of data points per cryptocurrency

Traditional optimization algorithms only required the Adjusted Close price. However, IA-based optimization requires additional information, so a new data frame with the Close price, High, Low, and Volume is required[16]. Furthermore, the DRL method requires splitting the data frame. 70% of the data was set aside for training and 30% for testing. As a result, the precise date between the two samples is June 13, 2021. Figure 1 in the appendix B depicts the adjusted closing prices for each cryptocurrency. And appendix B tables 4 and 5 show the structure of the data set used for both approaches.

4 Results (insides)

This section will deploy the gathered findings after executing the hold-created Python code. The section structure will give the outcomes of traditional approaches first, then DRL

likelihood of moving from one state to another, given a previously performed activity. The goal is to determine what action to take in each stage to maximize an expected payoff.

[16] The highest price represents the highest price of the stock traded that day, the lowest price represents the lowest price of the stock exchanged that day, and the volume represents the number of shares exchanged that day.

algorithms, and lastly, a comparison of both techniques.

4.1 Traditional approaches

Each optimization problem received an extra constraint to minimize nonzero weights. Portfolio optimization, as mentioned in earlier chapters, aims to have the best combination of assets feasible to achieve the desired results; nevertheless, it is fundamental to note that even with this constraint, some crypto assets still do not have weights for some of the portfolios.

4.1.1 Markowitz portfolio

Table 1 shows the outcomes of this optimization technique. Following this optimization technique with the chosen circumstances, an investor may achieve a 0.30 yearly return with a risk reflected by the volatility indicator of 0.77. Sharpe's ratio is 0.34, which means that for every 1 unit of risk, investors can expect to earn 0.34 units of return.

Table 1. Markowitz portfolio optimal weights

	BTC	LTC	PPC	DOGE	GRC	XPM	XRP	NXT	DASH	NEO	XMR	XVG	XLM	VTC	ETH	ETC	USDT	FIRO
Weights	11.5%	5.4%	1.3%	25.5%	0.0%	0.6%	8.7%	0.0%	0.0%	0.0%	6.6%	0.5%	9.4%	0.0%	13.4%	7.2%	10.0%	0.0%

```
Expected annual return = 0.30
Annual volatility = 0.77
Sharpe Ratio = 0.34
```

4.1.2 Minimum volatility portfolio

Table 2 shows the outcomes of this optimization technique. Following this optimization technique with the chosen circumstances, an investor may achieve a -0.02 yearly return with a risk reflected by the volatility indicator of 0.67. Sharpe's ratio is -0.09, which means that the portfolio has lost money and has underperformed the risk-free investment.

Table 2. Minimum Volatility portfolio optimal weights

	BTC	LTC	PPC	DOGE	GRC	XPM	XRP	NXT	DASH	NEO	XMR	XVG	XLM	VTC	ETH	ETC	USDT	FIRO
Weights	6.9%	5.6%	6.1%	4.5%	4.5%	2.9%	5.5%	6.0%	5.5%	5.1%	5.8%	4.3%	5.3%	5.8%	5.9%	5.2%	10.4%	4.9%

```
Expected annual return = -0.02
Annual volatility = 0.67
Sharpe Ratio = -0.09
```

4.1.3 Tangency portfolio

Table 3 shows the outcomes of this optimization technique. Following this optimization technique with the chosen circumstances, an investor may achieve a 0.60 yearly return with a

risk reflected by the volatility indicator of 1.22. Sharpe's ratio is 0.47, which means that for every 1 unit of risk, investors can expect to earn 0.47 units of return.

Table 3. Tangency portfolio optimal weights

	BTC	LTC	PPC	DOGE	GRC	XPM	XRP	NXT	DASH	NEO	XMR	XVG	XLM	VTC	ETH	ETC	USDT	FIRO
Weights	9.7%	0.0%	0.0%	60.9%	0.0%	0.0%	4.4%	0.0%	0.0%	0.0%	0.0%	0.0%	7.1%	0.0%	17.8%	0.1%	0.0%	0.0%

```
Expected annual return = 0.60
Annual volatility = 1.22
Sharpe Ratio = 0.47
```

4.1.4 Equally weighted portfolio

Table 4 shows the outcomes of this optimization technique. Following this optimization technique with the chosen circumstances, an investor may achieve a 0.23 yearly return with a risk reflected by the volatility indicator of 0.53. Sharpe's ratio is 0.27, which means that for every 1 unit of risk, investors can expect to earn 0.27 units of return.

Table 4. Equally weighted portfolio optimal weights

	BTC	LTC	PPC	DOGE	GRC	XPM	XRP	NXT	DASH	NEO	XMR	XVG	XLM	VTC	ETH	ETC	USDT	FIRO
Weights	5.6%	5.6%	5.6%	5.6%	5.6%	5.6%	5.6%	5.6%	5.6%	5.6%	5.6%	5.6%	5.6%	5.6%	5.6%	5.6%	5.6%	5.6%

```
Expected annual return = 0.23
Annual volatility = 0.53
Sharpe Ratio = 0.27
```

Appendix B figure 2 depicts a bar plot with optimal weights for the traditional approaches. However, in this case, the reader can see how the tangency portfolio provides higher expected annual returns than the other approaches. An explanation could be that the tangency portfolio puts more weight on cryptocurrencies with higher expected returns. But, it is also worth noticing that it has the highest annual volatility since it is weighted heavily against riskier assets.

4.2 IA-based approaches

With a learning rate of 0.0002, the total number of time steps assigned to all models was between 40.000 and 50.000. In addition, the algorithms consider an initial cash amount of 1M, a transaction cost of 0 (since cryptocurrency transactions are either extremely cheap or almost free), and a reward scale for the agent of 1e-1.

4.2.1 A2C

Table 5 shows the outcomes of this optimization technique. Following this optimization technique with an initial investment of 1M the total end value of the portfolio will be near 5.4M. Sharpe's ratio is 0.87, which means that for every 1 unit of risk, investors can expect to earn 0.87 units of return.

Table 5. A2C after training outcomes

```
================================
begin_total_asset:1000000
end_total_asset:5465897.051094722
Sharpe:  0.8711983602980227
================================
```

4.2.2 PPO

Table 6 shows the outcomes of this optimization technique. Following this optimization technique with an initial investment of 1M the total end value of the portfolio will be near 4.4M. Sharpe's ratio is 0.81, which means that for every 1 unit of risk, investors can expect to earn 0.81 units of return.

Table 6. PPO after training outcomes

```
================================
begin_total_asset:1000000
end_total_asset:4486599.640507633
Sharpe:  0.8115883922494571
================================
```

4.2.3 DDPG

Table 7 shows the outcomes of this optimization technique. Following this optimization technique with an initial investment of 1M the total end value of the portfolio will be near 8.7M. Sharpe's ratio is 1.015, which means that for every 1 unit of risk, investors can expect to earn 1.015 units of return.

Table 7. DDPG after training outcomes

```
================================
begin_total_asset:1000000
end_total_asset:8717186.206529703
Sharpe:  1.015597776491612
================================
```

It is possible to notice that such results are excessive, that the initial investment is nearly six-folded or even more, but this is not always the case, and there is an explanation for this situation. During the beginning of the testing period, the crypto market saw a boom, with each crypto asset's return reaching its historical highs; hence, the DRL techniques are considering a bullish scenario, which might explain why the initial investment value was almost six-folded. Furthermore, many institutional investors are becoming interested in this industry, boosting the crypto market (Sensoy and Akdeniz 2022).

It is vital to note that different strategies may assign more weight to specific assets based on

prior performance or other factors, in this case, trying to obtain a better Sharpe ratio after each time step. Moreover, some algorithms also consider the allocation of the weights based on the expected return of each asset.

4.3 Traditional approaches VS IA-based approaches

A graph displaying the cumulative returns for all the approaches is provided. This figure was applied exclusively to the test sample, which implies it only covers 30% of the total data set. In figure 6, one can see that all DRL approaches outperform the traditional approaches. Premise fulfilled for all of the IA-based models, except for the PPO algorithm in which the Equally Weighted portfolio achieved a higher cumulative return; one explanation for this portfolio outperforming the PPO approach is that the Equally Weighted portfolio ensures that each asset contributes equally to the overall performance. As a result, this can assist in minimizing the overall risk of the portfolio while still producing significant returns.

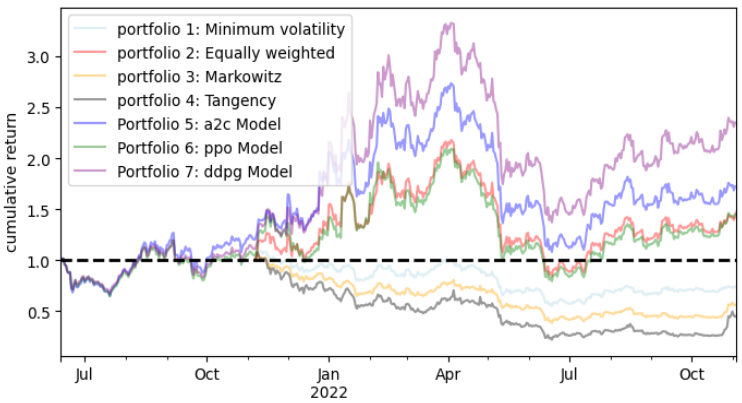


Figure 6. Cumulative returns for the testing period (13/06/2021-04/11/2022)

Table 8 displays the principal financial indicators of the attained findings. The results for each approach may differ from the previously presented ones since now they all are applied just to the testing data. Still, the study's premise continues to be fulfilled since DRL algorithms outperform traditional techniques in terms of returns and volatility, as well as the Sharpe ratio[17], which the reader is already familiar with.

Table 8. Principal performance metrics comparing each approach results

	Minimum volatility	Equally weighted	Markowitz	Tangency	a2c Model	ppo Model	ddpg Model
Annual return	-0.177106	0.307904	-0.321035	-0.418491	0.310031	0.202259	0.526939
Cumulative returns	-0.245980	0.475197	-0.429251	-0.543984	0.725402	0.450720	1.351207
Annual volatility	0.364275	0.688524	0.443797	0.718395	0.699144	0.695682	0.712967
Sharpe ratio	-0.353025	0.730328	-0.651233	-0.399841	0.732796	0.607534	0.942172

[17] Table 8 shows the best-performing strategy for each indicator in green, the second-best-performing strategy in yellow, and the worst-performing strategy in red.

5 Conclusion

To sum up, after testing numerous approaches for the portfolio optimization problem applied to the highly volatile environment in which the cryptocurrency world evolves, the DRL models outperform traditional methods in terms of not only the Sharpe ratio but also risk-adjusted return. Meanwhile, traditional methods are only concerned with maximizing portfolio returns, while more advanced strategies can take more inputs, which helps to account for high volatility and thus achieve better results, and the way those models are trained through trial and error to achieve better results also helps in the pursuit of better results. In terms of cumulative returns, the DDPG algorithm outperformed all other techniques, owing to its ability to handle continuous action spaces to discover the best strategy significant in portfolio optimization problems with an infinite number of alternative actions.

Traditional techniques perform better in terms of computational efficiency for the portfolio optimization problem since they do not work on an iteration basis, which consumes a lot of computational resources.

The research for this working project used various fake assumptions that do not adequately represent a real-world trade environment. Short-selling and hedging losses, for example, were not considered in the research. As a result, future research on the issue can incorporate such qualities into the algorithm; Furthermore, in a bearish circumstance, a test can be performed to observe how the DRL agents perform. Furthermore, because the circulating supply of each crypto asset is significantly related to its volatility, the model should consider this information; regrettably, in this situation, that information could not be added since no publicly available source contains it daily.

All of the work done in this working project is not intended to be and should not be interpreted as financial advice to engage in cryptocurrency transactions or to be incorporated as part of an investment strategy.

References:

Achiam, Joshua. 2020. “Spinning Up Documentation Release.”

Arulkumaran, Kai, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. 2017. “A Brief Survey of Deep Reinforcement Learning”

Badea, Leonardo. 2008. “Determining the Efficiency Frontier.”

Benhamou, Eric, David Saltiel, Jean Jacques Ohana, Jamal Atif, and Rida Laraki. 2020. “Deep Reinforcement Learning (DRL) for Portfolio Allocation”

Bodie, Zvvi, Alex Kane, and Alan J Marcus. 2014. “Investments.”

Ceria, Sebastián, and Kartik Krishnan Sivaramakrishnan. 2013. “Portfolio Optimization.”

Cotta, C., E. Alba, R. Sagarna, and P. Larrañaga. 2002. “Adjusting Weights in Artificial Neural Networks Using Evolutionary Algorithms.”

Cy, and Polyviou. 2020. “Advantages and Risks Associated with Portfolio Maximization.”

Dapp, Marcus M, Dirk Helbing, and Stefan Klauser. 2021. “From Fiat to Crypto: The Present and Future of Money.”

Degris, Thomas, Patrick M. Pilarski, and Richard S. Sutton. 2012. “Model-Free Reinforcement Learning with Continuous Action in Practice.” In *Proceedings of the American Control Conference*, 2177–82. Institute of Electrical and Electronics Engineers Inc.

Doumenis, Yianni, Javad Izadi, Pradeep Dhamdhere, Epameinondas Katsikas, and Dimitrios Koufopoulos. 2021. “A Critical Analysis of Volatility Surprise in Bitcoin Cryptocurrency and Other Financial Assets.”

Durall, Ricard. 2022. “Asset Allocation: From Markowitz to Deep Reinforcement Learning.”

FinRl. 2022. “FinRL Documentation Release.”

FRED. 2022. “3-Month Treasury Bill Secondary Market Rate, Discount Basis (TB3MS).”

Gemechu, Tekle. 2020. “Nonlinear Problem (System) Analysis.”

Houben, Robby, and Alexander Snyers. 2018. “Cryptocurrencies and Blockchain.”

Jin, Mingzhou, Zexin Li, and Shengkai Yuan. 2021. “Research and Analysis on Markowitz Model and Index Model of Portfolio Selection.”

Karandikar, Rajeeva L. 2019. “Modelling in the Spirit of Markowitz Portfolio Theory in a Non-Gaussian World.”

Li, Yuming, Pin Ni, and Victor Chang. n.d. “Application of Deep Reinforcement Learning in Stock Trading Strategies and Stock Forecasting.”

Lillicrap, Timothy P., Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. “Continuous Control with Deep Reinforcement Learning.”

Mangram, Myles, and Myles E Mangram. 2013. “A Simplified Perspective of the Markowitz Portfolio Theory.”

Milthaler, Frank. 2020. “FinQuant Documentation Release.”

Mnih, Volodymyr, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. “Asynchronous Methods for Deep Reinforcement Learning.”

Omisore, Iyiola. 2012. “The Modern Portfolio Theory as an Investment Decision Tool.” *Journal of Accounting and Taxation* 4

Sadriu, Lorik. 2022. “Deep Reinforcement Learning Approach to Portfolio Optimization.”
Schmid, Friedrich, and Rafael Schmidt. 2010. “Statistical Inference for Sharpe Ratio.” In *Interest Rate Models, Asset Allocation and Quantitative Techniques for Central Banks and Sovereign Wealth Funds*, 337–57. Palgrave Macmillan UK.
Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. “Proximal Policy Optimization Algorithms.”
Sensoy, Ahmet, and Levent Akdeniz. 2022. “Retail vs Institutional Investor Attention in the Cryptocurrency Market.”
StableBaselines3. 2022. “Stable Baselines3 Documentation.”
Sutton, Richard S, and Andrew G Barto. 2017. “Reinforcement Learning:An Introduction.”
Szepesvári, Csaba, Amir Massoud Farahmand, Azad Shademan, and Martin Jägersand. 2009. “Model-Based and Model-Free Reinforcement Learning for Visual Servoing.”
Yang, Hongyang, Xiao-Yang Liu, Shan Zhong, and Anwar Walid. n.d. “Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy.”
Zivot, Eric. 2021. “Introduction to Computational Finance and Financial Econometrics with R.”

Appendix A

Appendix A complements what is shaped in Section 1.4 by demonstrating an example of an N assets portfolio optimization problem using matrix notation.

First, let $R_i = i = (1, 2 \dots, n)$ denote each asset return, and w_i denotes the slice of the capital invested in each asset i. Thus, using matrix algebra, one can create an $nx1$ vector depicting the asset returns and weights, as seen below.

$$R = \begin{pmatrix} R_1 \\ R_2 \\ \vdots \\ R_n \end{pmatrix};\ w = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix}$$

To get the overall portfolio return, multiplying the first vector by the transpose of the second vector is needed.

$$R_{p,w} = R * w^T$$

A nx1 matrix holding the historical data of the returns, commonly known as the mean of the returns, may be constructed using the previously supplied vectors.

$$E[R] = E\begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_n \end{bmatrix} = \begin{pmatrix} E[R_1] \\ E[R_2] \\ \vdots \\ E[R_n] \end{pmatrix} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{pmatrix} = \mu$$

And thus, the expected returns of the portfolio can be depicted as follows:

$$\mu_{p,w} = E[w^T R] = w^T \mu = (w_1, w_2, \dots, w_n) \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{pmatrix}$$

Below is the covariance matrix of the returns:

$$Var(R) = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{1n} \\ \sigma_{21} & \sigma_2^2 & \sigma_{2n} \\ \sigma_{3n} & \sigma_{32} & \sigma_n^2 \end{pmatrix} = \sum.$$

And thus, the covariance matrix of the portfolio is represented by the following equation:

$$\sigma_{p,w}^2 = var(w^T R) = w^T \sum w = (w_1, w_2, \dots, w_n) \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{1n} \\ \sigma_{21} & \sigma_2^2 & \sigma_{2n} \\ \sigma_{3n} & \sigma_{32} & \sigma_n^2 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix}$$

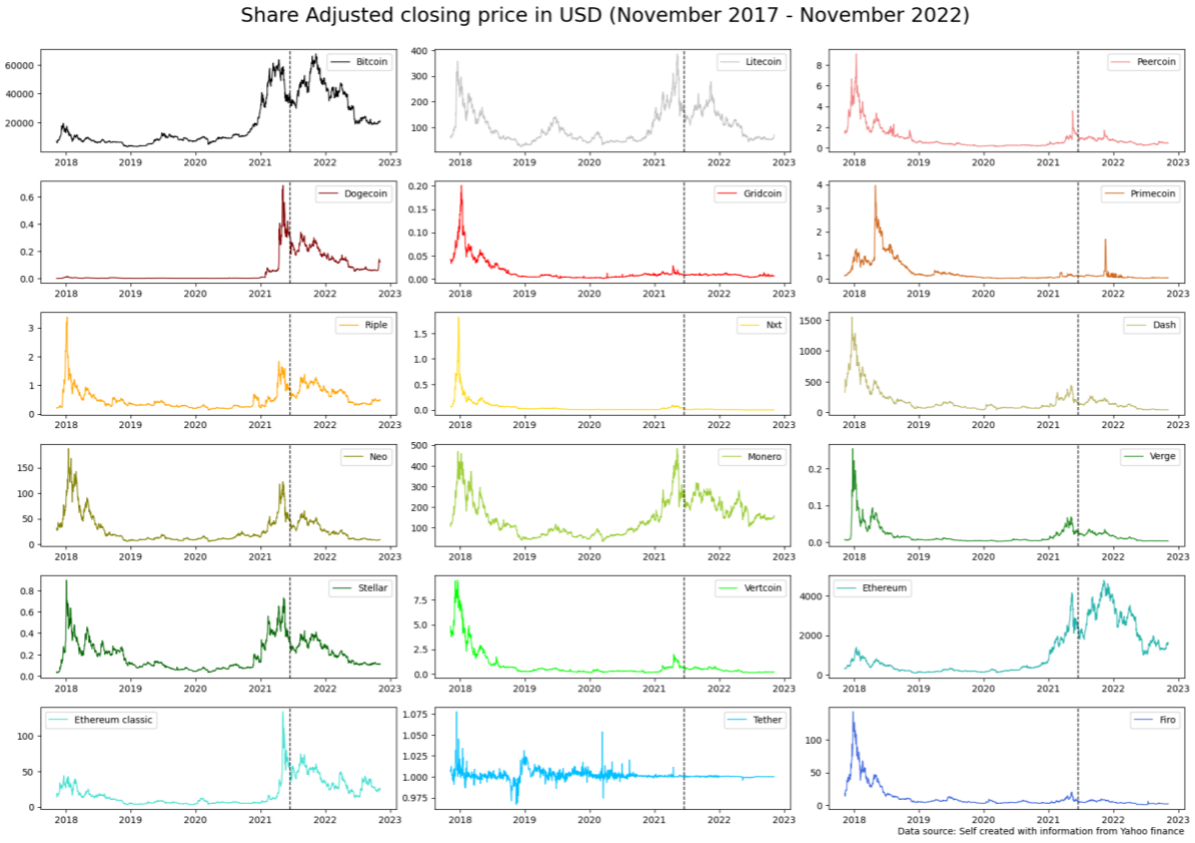
It is also worth noting that for the Markowitz optimization problem, the total of all weights must equal one; this criterion is stated by the following equation:

$$w^T 1 = (w_a, w_b, \dots, w_n)\begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} = w_a + w_b + \cdots + w_n = 1$$

The matrixial equations structure were adapted from (Zivot 2021).

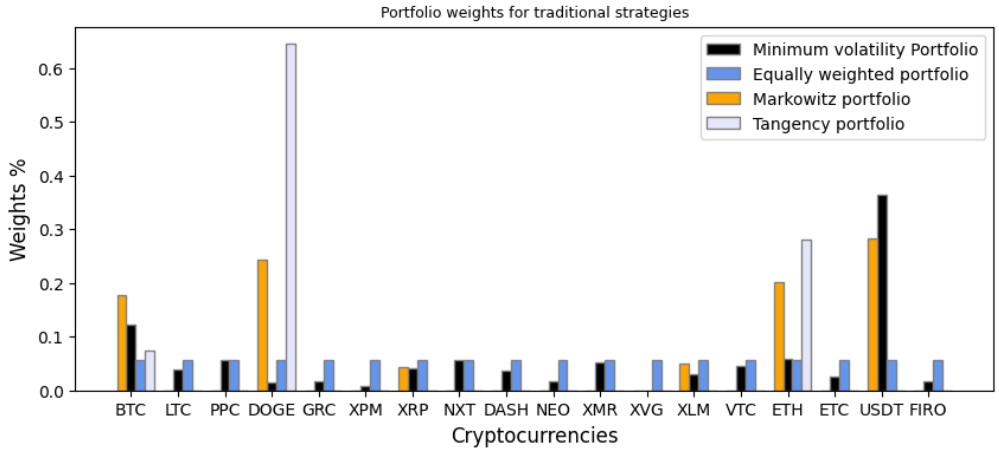
Appendix B

Figure 1 shows the adjusted closure prices for all the crypto assets chosen for the hold period. Each picture includes a vertical dotted line to indicate the exact date (June 13, 2021) when the data split into training and testing samples occur.



Appendix figure 1. Share adjusted closing price in USD

Figure 2 shows the optimal weights for each of the conventional approaches.



Appendix figure 2. Portfolio weights

Table 1 shows an example of a Q-table with n actions and n states and their corresponding q values.

Appendix table 1. Q-table example

Actions

States

	a_1	a_2	a_3	a_4	a_n
s_1	3	16	11	10	17
s_2	19	8	-10	8	-5
s_3	0	17	-1	-9	1
s_4	2	24	27	-9	-7
s_n	1	-6	-8	14	30

Table 2 shows the financial indicators utilized to feed the AI-based algorithms. Also, there is a definition for each metric in table two.

Appendix table 2. Used Financial indicators for the AI models

Financial indicators	Definition
Annual Relative Strength Index (RSI)*	Compares the number of recent gains and losses over a specific period to assess the speed and change of a security's price movements.
Chaikin Money Flow (CMF)*	CMF indicator measures the amount of money flowing into and out of an investment.
Average True Range (ATR)*	ATR is a technical indicator that monitors a security's volatility.
Simple Moving Average (SMA)*	SMA is a technical indicator used to smooth out price activity to better discern the underlying trend's direction.
Covariance matrix**	The covariance matrix is a square matrix that calculates the variation among each asset in the matrix and every other asset in the matrix.
Turbulance index***	The turbulence index measures the overall volatility of the stock market.

* Definitions adapted from the Python Technical Analysis Library package
** Definition extracted from Janakiev Nikolai. (2018, August 3). "Understanding the Covariance Matrix"
*** Definitions extracted from the Python Portfolio Optimizer Library package

Table 3 shows the descriptive statistics for the stock pool chosen for the study period.

Appendix table 3. descriptive statistics of all cryptocurrencies

Symbols	count	mean	std	min	25%	50%	75%	max
BTC	1823.0	20208.052572	17058.194033	3236.761719	7624.915039	10760.066406	33772.007812	67566.828125
LTC	1823.0	102.333945	64.317799	23.464331	53.660265	76.513245	140.225647	386.450775
PPC	1823.0	0.883302	1.057644	0.111957	0.289076	0.512863	0.972825	9.053410
DOGE	1823.0	0.059978	0.099713	0.001038	0.002583	0.003591	0.070007	0.684777
GRC	1823.0	0.015099	0.022655	0.001370	0.004710	0.008510	0.010875	0.200795
XPM	1823.0	0.275378	0.457358	0.015766	0.041781	0.109486	0.228795	3.968930
XRP	1823.0	0.528895	0.369089	0.139635	0.280632	0.395380	0.694685	3.377810
NXT	1823.0	0.056147	0.136082	0.002824	0.010324	0.015703	0.042036	1.817110
DASH	1823.0	177.738458	194.155138	39.720448	73.244160	109.650917	192.029167	1550.849976
NEO	1823.0	27.910954	27.596266	5.377224	9.958423	17.040951	37.468685	187.404999
XMR	1823.0	149.517743	88.703141	33.010323	72.297558	129.581436	209.433846	483.583618
XVG	1823.0	0.017944	0.025472	0.001868	0.004486	0.008055	0.022286	0.255441
XLM	1823.0	0.194010	0.136995	0.028182	0.083735	0.135778	0.270109	0.896227
VTC	1823.0	0.827832	1.376053	0.090590	0.243826	0.351730	0.664622	9.521380
ETH	1823.0	1126.550386	1205.811373	84.308296	208.786240	472.902008	1804.498718	4812.087402
ETC	1823.0	19.829790	18.345207	3.472387	6.041255	12.062400	29.510000	134.101791
USDT	1823.0	1.001633	0.005786	0.966644	0.999993	1.000513	1.002550	1.077880
FIRO	1823.0	11.449173	17.292159	1.257650	4.047136	5.593007	9.210330	142.434006

Table 4 shows the structure of the first five rows of the data set utilized for the traditional approaches methods.

Appendix table 4. Structure of traditional approach data frame

Symbols	BTC	LTC	PPC	DOGE	GRC	XPM	XRP	NXT	DASH	NEO	XMR	XVG	XLM	VTC	ETH	ETC	USDT
Date																	
2017-11-09	7143.580078	64.269699	1.64380	0.001415	0.042891	0.141829	0.217488	0.071503	326.007996	31.903200	120.779999	0.007463	0.039946	4.83677	320.884003	14.2095	1.00818
2017-11-10	6618.140137	59.260101	1.42203	0.001163	0.039688	0.135370	0.206483	0.064071	329.571014	28.178900	105.585999	0.006466	0.033073	4.19009	299.252991	14.6031	1.00601
2017-11-11	6357.600098	62.303299	1.43993	0.001201	0.038134	0.132312	0.210430	0.068137	346.056000	28.528099	119.615997	0.006238	0.033053	4.32257	314.681000	19.4209	1.00899
2017-11-12	5950.069824	59.005402	1.40257	0.001038	0.033628	0.114648	0.197339	0.060480	536.116028	26.961700	123.856003	0.005454	0.028182	3.75240	307.907990	15.1837	1.01247
2017-11-13	6559.490234	61.396500	1.48639	0.001211	0.036706	0.164880	0.203442	0.063419	427.372986	28.402399	123.402000	0.006124	0.030656	4.45275	316.716003	16.1059	1.00935

Table 5 shows the structure of the first rows of the data set utilized for the IA methods.

Appendix table 5. Structure of AI approach data frame

	date	tic	close	high	low	volume
0	2017-11-09	BTC-USD	7143.580078	7446.830078	7101.520020	3226249984
1	2017-11-09	LTC-USD	64.269699	66.838303	61.725800	279652992
2	2017-11-09	PPC-USD	1.643800	1.688590	1.528690	723863
3	2017-11-09	DOGE-USD	0.001415	0.001415	0.001181	6259550
4	2017-11-09	GRC-USD	0.042891	0.044740	0.041436	161914
...	...	...	...	...	...	...
32791	2022-11-04	VTC-USD	0.185065	0.187668	0.178969	91076
32792	2022-11-04	ETH-USD	1645.152710	1661.334717	1530.043335	20693954560
32793	2022-11-04	ETC-USD	25.541330	25.963387	23.974455	548248128
32794	2022-11-04	USDT-USD	1.000060	1.000117	0.999977	87140155392
32795	2022-11-04	FIRO-USD	2.311784	2.376280	2.226519	3425163

32796 rows × 6 columns