UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Deep Reinforcement Learning and Signal Processing applications for Investment Strategies.

A Degree Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Victor Estrada Nájar

In partial fulfilment

of the requirements for the degree in

TELECOMMUNICATIONS TECHNOLOGIES AND SERVICES ENGINEERING

Advisor: JOSEP VIDAL

Barcelona, February 2021

Abstract

In this project we look at the fundamentals of Finance, Deep Reinforcement Learning and signal Processing in order to develop an investment strategy for the stock market.

The study parts from a development made by another university in which an Ensemble technique is designed using three different DRL algorithms (A2C, DDPG and PPO)

In our case, the objective is to improve the very promising results obtained by the author.

Three possible improvements are proposed, such as using the Differential Sharpe Ratio as a reward function, expanding the database to another containing a broader universe of financial assets and finally it is proposed to carry out a combination strategy with all three algorithms using signal processing techniques.

After exploring the technical difficulties and proposing formal solutions, it is demonstrated that in all three cases performance is improved and results are compared to the previous ones.

Resum

En aquest projecte fem una ullada als fonaments de les Finances, del Reinforcement Learning i del Processat de Senyal per a desenvolupar una estratègia d'inversió als mercats financers.

L'estudi parteix d'un desenvolupament ja fet per una altra universitat en el que es dissenya una tècnica d'acoblament amb tres algorismes de DRL (A2C,PPO i DDPG)

En el nostre cas, l'objectiu es millorar els prometedors resultats obtinguts per l'autor.

Tres possibles solucions s'expliquen al projecte, la primera fer servir el Sharpe Ratio Diferencial com a funció de reward dels Agents DRL. La segona fer servir una base de dades més gran amb un univers d'accions més extens. Per últim es proposa una tècnica de combinació dels algorismes de DRL basada en processat de senyal.

Després de discutir les dificultats tècniques i de fer les propostes formals es demostra en els tres casos la millora de rendiment i es comparen els resultats amb la estratègia original.

Resumen

En este proyecto miramos los fundamentos de las Finanzas, del Deep Reinforcement Larning, y del procesado de señal para construir una estrategia de inversión en bolsa.

El estudio parte de un desarrollo hecho por otra universidad en la que se estudia una técnica de ensamblado usando tres algoritmos de DRL (A2C, DDPG y PPO).

En nuestro caso el objetivo reside en mejorar los muy prometedores y ambiciosos resultados obtenidos por el anterior académico.

Se proponen tres posibles mejoras, como cambiar la función de Reward de los Agentes de DRL al Sharpe Ratio, se propone ensanchar la base de datos a otra que contenga un universo de activos financieros más amplio y por último se propone realizar una estrategia de combinado para los tres algoritmos, usando técnicas de procesado de señal.

Tras explorar las dificultades técnicas y proponer soluciones formales, se demuestra en los tres casos que se consigue mejorar el rendimiento de la estrategia original y se compara en todo momento con los anteriores resultados.

Acknowledgements

I would like to thank Josep Vidal and Montserrat Najar for accepting my project, giving me the opportunity to develop my thesis in sync with my interests for Finance, and helping me during the ideation and development phase.

I would like to thank my Family and Nina Blasi, my couple, for being my emotional support during this time.

And finally, I would like to thank all my friends who made late night study sessions more bearable.

To you all, Thank you.

Revision history and approval record

Revision	Date	Purpose
0	10/01/2021	Document creation
1	20/01/2021	Document revision

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Víctor Estrada Nájar	victor.estrada.najar@gmail.com
Josep Vidal Manzano	Josep.vidal@upc.edu
[Project Supervisor 2]	

Written by: Victor Estrada Nájar		Reviewed and approved by:Josep Vidal Manzano	
Date	10/01/2021	Date	20/01/2021
Name	Victor Estrada Nájar	Name	Jospe Vidal Manzano
Position	Project Author	Position	Project Supervisor

Table of contents

List of Figures

List of Tables:

1. Introduction

1.1. Motivation

Imagine a global system in which millions of people interact daily with each other whit the sole goal of making a profit. The statistical analysis of financial markets and its dynamics added to the influence the human factor has in them make financial markets a very complex yet interesting field. Understanding these dynamic and modelling them in a quantitative way has been a goal of mine for years, and this project was the absolute best opportunity to do it.

The idea of building a bridge between Machine learning and portfolio management comes from my own interest and passion for financial matters, investments and asset management, which I have been studying personally on my own for several years.

From the very beginning I wanted to study finance from a more quantitative perspective and I had it very clear in my head that in my final degree thesis I would indeed try to apply the tools I learnt during my Telecommunications Engineering degree in finance.

After having researched in the fields of interest, I discovered a whole lot of signal processing and deep learning technics that could make for a great project.

I decided to focus into Trading and Asset Management. I discovered that in those fields there was a very new approach by using machine learning, since many of the traditional techniques, based on models, that are used currently do not provide good enough returns. Research in the application of deep reinforcement learning in asset management has shown significant gains in terms of profit.

1.2. Project objectives

The project main goals were:

- Get an understanding in Deep Reinforcement Learning techniques and Neural Networks.
- Get an understanding in Financial Markets Dynamics, Portfolio Management theory and Asset Management.
- Review the work that has been carried by other researchers and institutions relating both fields.
- Download financial data from all the assets taken into account (S&P500) from public databases such as yahoo finance or google finance and learn how to process it.
- Propose one or several investment strategies using both asset management tools and DRL technics.
- Develop techniques which improve current studies and results.
- Train, test and simulate the developments in order to check its viability and profitability.

1.3. Project requirements:

- Correct downloading of Financial Market Data, more specifically from all of the stocks conforming the S&P500.
- Preprocessing of financial data into data frames to calculate all the parameters and statistics needed for the analysis.
- Evaluate the most effective way to choose from our stock's universe to invest in portfolio.
- Carefully evaluate and choose a set of factors that provide real market conditions for our models.
- Training the models.
- Choosing the most effective algorithm in each scenario.

1.4. Project specifications

The performance needed by our software has to match at least the markets returns (DJI). A good performance will be considered if our returns are higher than the markets one. An outstanding performance will be considered if the returns obtained are higher than the ones stated in previous papers where DRL is applied in finance.

1.5. Workplan

1.5.1. Work Packages

<table>
<tr><td colspan="2">Project: TFG Victor Estrada</td><td colspan="2">WP ref: 1</td></tr>
<tr><td colspan="2">Major constituent: Learning and research phase</td><td colspan="2">Sheet 1 of 1</td></tr>
<tr><td colspan="2" rowspan="2">Short description: In this first phase we plan to recap as much information as possible in regards of the topics related to the project such as deep reinforcement learning, financial markets, portfolio management and trading, signal processing and quantitative finance.

That exhaustive due diligence will be carried in order to see what has been done in the academic field regarding our line of research.</td><td colspan="2">Planned start date: 21-09-2020
Planned end date: 21-10-2020</td></tr>
<tr><td colspan="2">Start event: 21-09-2020
End event: 21-10-2020</td></tr>
<tr><td>Internal task T1: Research on Deep Reinforcement Learning

Internal task T2: Portfolio management learning

Internal task T3: Research on the use of RL in portfolio management</td><td>Deliverables:</td><td>Dates:</td></tr>
</table>

<table>
<tr><td>Project: TFG Victor Estrada</td><td colspan="2">WP ref: 2</td></tr>
<tr><td>Major constituent: Testing</td><td colspan="2">Sheet 1 of 1</td></tr>
<tr><td rowspan="2">Short description: The aim of the second workpackage is to build the adequate investment strategy and develop the according software in order to simulate the strategy.

For that we will focus on this phase in testing different proposed ideas in order to see what can be improved in the already developed solutions by other researchers.</td><td colspan="2">Planned start date: 22-10-2020
Planned end date: 03-12-2020</td></tr>
<tr><td colspan="2">Start event: 22-10-2020
End event: 03-12-2020</td></tr>
<tr><td>Internal task T1: Using a bigger database

Internal task T2: Modifying parameters such as training and validation windows in the models

Internal task T3: Change reward function from absolute return to Sharpe ratio

Internal task T4: Using other features and indicators

Internal task T5: Using other criteria to choose investment strategy apart from Sharpe ratio such as Jensen Alpha, Beta, R-Squared…</td><td>Deliverables:</td><td>Dates:</td></tr>
</table>

<table>
<tr><td>Project: TFG Victor Estrada</td><td colspan="2">WP ref: 3</td></tr>
<tr><td>Major constituent: Solution Development</td><td colspan="2">Sheet 1 of 1</td></tr>
<tr><td rowspan="2">Short description: In this phase all the results obtained previously in the testing phase will be evaluated and added carefully in the already developed program to try and improve the Ensambling Strategy.</td><td colspan="2">Planned start date:4-12-2020
Planned end date:11-01-2021</td></tr>
<tr><td colspan="2">Start event: 4-12-2020
End event: 11-01-2021</td></tr>
<tr><td></td><td>Deliverables:</td><td>Dates:</td></tr>
</table>

<table>
<tr><td>Project: TFG Victor Estrada</td><td>WP ref: 4</td></tr>
<tr><td>Major constituent: Validation and Simulation</td><td>Sheet 1 of 1</td></tr>
<tr><td rowspan="2">Short description: The aim of this work package is to Validate whether some improvements where possible or not and Simulate the obtained trading results using our new model compared against older ones and industry benchmarks.</td><td>Planned start date:04-01-2021
Planned end date:11-01-2021</td></tr>
<tr><td>Start event: 04-01-2021
End event: 11-01-2021</td></tr>
</table>

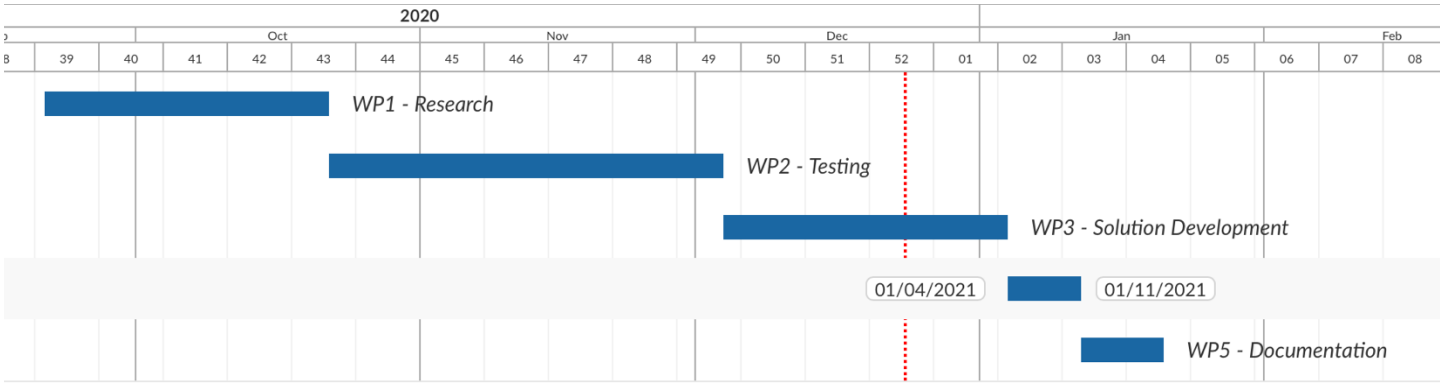
Project: TFG Victor Estrada	WP ref: 5	
Major constituent: Documentation	Sheet 1 of 1	
Short description: In this final phase all the results and the process will be documented, as well as all the learnings obtained during the project.	Planned start date:12-01-2021 Planned end date:20-01-2021	
	Start event: 12-01-2021 End event: 20-01-2021	
	Deliverables: Final Degree Thesis	Dates:24-01-21

1.5.2. Project Milestones

WP#	Task#	Short title	Milestone / deliverable	Date (week)
1	1	Recap research studies	Come up with as many articles on the topic as can be found. Get ourselves used on the financial applications of DRL and signal processing. Benchmarking	10-10-2020
2	1	Download needed data	Scrapping from the internet the financial Data that will be used. The data will be the stocks conforming the S&P500 in the determined timeframe.	10-11-2020
2	2	Successfully run the basic form of the program	Being able to run the code that has been developed prior to ours in some research studies and which is public for use. We will use it as a starting point of development and try to improve it by making changes or adding new functionalities.	25-11-2020
3	1	Come up with possible improvements	In order to improve the results a list of possible improvements or weak points has to be developed.	08-12-2020
3	2	Test weak points and improvements	Our hypothesis formulated in the previous part have to be tested.	12-12-2020
4	1	Add Various functionalities (Algorithms, pre-selection	Add to the starting code new functions, or changes and be able to run all of them seamlessly.	18-12-2020

		techniques, contingencies…)		
4	2	Train and Simulate the software on market condition	Once the changes have been made and the best overall form has been determined, properly run, train and test our models under real data and real market constraints to see what returns we can achieve.	6-01-2021
5	1	Document	Document and iterate the process until best results are achieved.	15-01-2021

1.5.3. Gantt Diagram



1.6. Deviations from original plan

The first idea of project we came up with had to do with Statistical Arbitrage. This approach was to look into statistical arbitrage and the relationship between financial markets and Covid-19 data. Later on, I found a very poor close to non-existent dependance on both, so the idea was discarded. This is why the project itself was delayed. Coming up with the final idea on what we would be working on took some extra time.

2. State of the art

Stock trading is used in investment companies, banks and hedge funds in order to maximize investment returns. Profitable capital allocation is fundamental, therefore the research in these fields and efforts to develop new technologies and winning strategies are immense. Analysts try to come up with new ideas to model the markets and take and edge. However, it is very challenging to quantify and consider all the factors that influence these very dynamic systems.

Portfolio management is an investment strategy that aims at maximizing the expected return on investment while minimizing risk by continuously reallocating the portfolio assets.

The existing works on Portfolio Management are not satisfactory, moreover due to the highly competitiveness of the field, firms make a lot of effort not to disclose their research.

The traditional approach to Portfolio Management relies on, firstly computing the expected stock return and the covariance matrix of stock prices. Then, the best portfolio allocation strategy is obtained by either maximizing the return for a given risk or minimizing the risk for a prespecified return. This approach, however, is complex and very costly to implement, although it is demonstrated that it outperforms the market, we would like better results.

Another approach for stock trading is to model it as a Markov Decision Process and use dynamic programming to derive the optimal strategy [1,2]. However, the scalability of this model is very limited.

In the recent years, machine learning and deep learning have been widely introduced in today's society, and financial markets have not been an exception. The first proposal to use Artificial Neural Networks for modelling the market was by [3].

Ever since, Applications have been developed to build prediction and classification models for the financial market. Fundamental Economic data, technical Statistical indicators and alternative data are combined with machine learning algorithms to extract new investment alphas and therefore have an edge on other investors. [4,5]. However, these approaches are only focused on picking stocks but not modeling positions.

A major break-through comes with the proposal of using deep reinforcement learning [6,7,8]

3. Financial Signal Processing

In this chapter the fundamental concepts needed to comprehend the project are presented. Also, the similarities between the fields of Finance and Signal Processing are addressed so the reader whom might be familiar with one of both can bridge their gaps to further understand the project.

Most of the financial applications with rigorous analysis rely on discrete data observations sorted chronologically over a set period of time, that is known as Time Series Analysis. Signal Processing, on the other hand, provides a rich toolbox for systematic time-series analysis, modelling and forecasting. Therefore, Signal Processing offers a wide range of available mathematical applications and algorithms which can be easily reinterpreted to be used with financial data [9,10].

3.1. Financial Terms and Concepts

3.1.1. Asset

An asset is a resource with an economic value that individuals, corporations or countries own or control with the expectation that it will provide a future profit.

There exists a wide variety of different asset classes out there: Current Assets, Fixed Assets, Intangible Assets and Financial Assets which includes securities such as equity or stocks and fixed income or bonds. For the scope of the project, we will go only into Financial Assets, more specifically stocks.

3.1.2. Portfolio

The concept of portfolio refers to a collection of multiple financial assets, described previously, held by an investor. A portfolio constitute by M assets can be characterized by vector $\mathbf{p} = [p_1, \quad \cdots \quad p_M]^T$, where p_i is the price of the *i-th* asset. The normalized amount invested at time t in each asset is defined by the portfolio vector $\mathbf{w}_t = [w_{1,t}, \quad \cdots \quad w_{M,t}]^T$.

Therefore, the value of the portfolio is defined as:

$$Portfolio\ Value = \boldsymbol{p}^T \mathbf{w}_t \qquad \text{and} \quad \mathbf{1}^T \mathbf{w}_t = 1 \tag{1}$$

A portfolio doesn't have to be static over time. Portfolios usually are managed by buying or selling assets in order to accumulate maximum returns and avoid risk. Portfolio management (i.e., the changes of accumulated stocks at each time) will be the center focus of this project.

3.2. Financial Time-Series

The statistical dynamics of financial markets, as a result of the non-static supply and demand balance of the Economy, causes prices to evolve over time. This makes financial data ideal to be modelled as a time series which provides the quantitative tools to extract useful (or predictable) information for future investments. These series are naturally discretized in days or months. Asset prices and returns can be modelled differently depending on the application they are needed for.

The investment profit or loss is defined by the returns that can be linear or simple return and log-return.

3.2.1. Simple Returns

The net or simple return represents the percentage change in asset prices during the holding time.

$$R_t \triangleq \frac{p_t - p_{t-1}}{p_{t-1}} = \frac{p_t}{p_{t-1}} - 1 \quad (2)$$

being p_t the price of an asset at time index t.

Then, the ratio between the end capital and the initial investment is named as the gross return:

$$1 + R_t = \frac{p_t}{p_{t-1}} \quad (3)$$

The cumulative gross return over k periods is the product of single period gross returns:

$$1 + R_t(k) = \prod_{i=0}^{k-1} \frac{p_{t-i}}{p_{t-i-1}} = \prod_{i=0}^{k-1}(1 + R_{t-i}) \quad (4)$$

and the corresponding net return is:

$$R_t(k) = \frac{p_t}{p_{t-k}} - 1 \quad (5)$$

3.2.2. Log-Returns

The log-return is defined as:

$$r_t \triangleq \log(1 + R_t) = \log \frac{p_t}{p_{t-1}} \quad (6)$$

The advantage of the log-returns is the additivity over periods, being the cumulative log-return over k periods:

$$r_t(k) \triangleq log\big(1 + R_t(k)\big) = log\left(\prod_{i=0}^{k-1}(1 + R_{t-i})\right) = \sum_{i=0}^{k-1} log(1 + R_{t-i}) = \sum_{i=0}^{k-1} r_{t-i} \quad (7)$$

3.2.3. Portfolio Returns

The net return of a portfolio composing of M assets over a single period t is

$$R_t^p = \mathbf{w}_t^T \mathbf{R}_t \quad (8)$$

being $\mathbf{R}_t = [R_{1t} \quad \cdots \quad R_{Mt}]$ is de vector with the simple returns of each asset. That means that the simple returns have the additive property in the portfolio domain where the log-returns lack of it. For this reason, dealing with portfolio management implies the preferred use of simple returns as shown in Figure 1.

3.3. Evaluation Criteria and Performance Metrics

The goal of all financial investment relies on designing an optimal portfolio under given characteristics.

For doing so, we need a set of statistical metrics and criteria which evaluate portfolios or investment strategies and provide a scenario from which we will be able to choose the ones which fit the best our needs.

The Risk Aversion Criterion is usually preferred by investors. That means choosing the optimal investment by gaining exposure to maximum return while lowering the exposure to minimum risk. In other words, if we were to have different portfolios with the same expected returns but with different volatilities, we would choose the one with lower volatility.

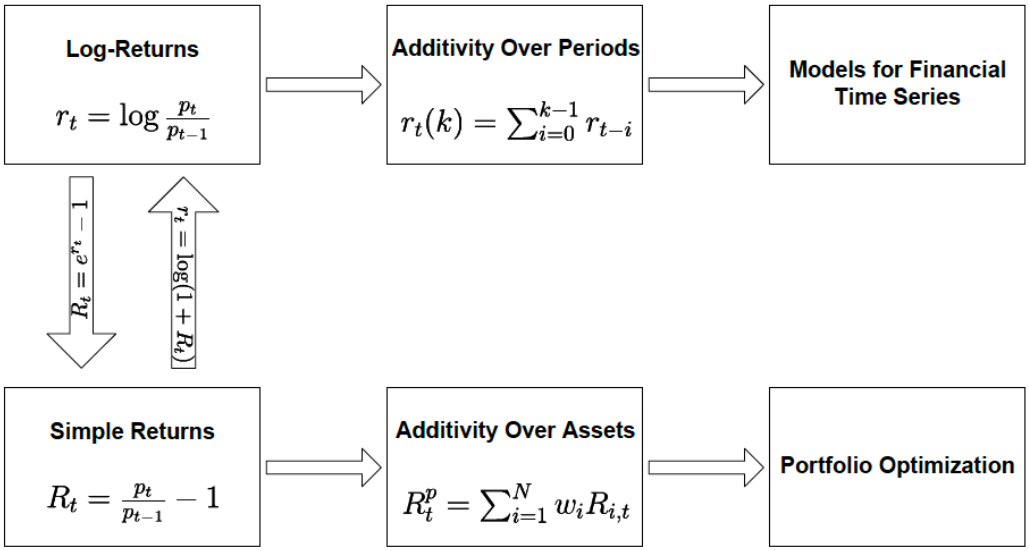


Figure1. Simple returns versus log-returns [10]

3.3.1. Cumulative Return (CR)

Cumulative Return (CR), defined in section 3.2, is the main performance criteria as indication of the total profit.

3.3.2. Sharpe Ratio (SR)

According to the Modern Portfolio Theory's focal point: Risk and reward must be evaluated together when considering investment choices. Here is where the Sharpe Ratio (SR) appears. SR is the measure of risk-adjusted return, defined as the ratio of the expected return of a given portfolio or investment strategy to its standard deviation and adjusted by a scaling factor.

$$SR_{1:T} = \sqrt{T}\ \frac{\mathbb{E}[\boldsymbol{r}_{1:T}]}{\sqrt{Var[\boldsymbol{r}_{1:T}]}} \tag{9}$$

Where T is the number of samples considered. (usually, T = 252 will be considered as a default, which is the total trading days there are in one year).

The analogue of the SR in signal processing would be the Signal to Noise Ratio (SNR) [10].

Higher SR value indicates better performance of portfolios subject to similar risk or similar reward subject to lower exposure to risk. Therefore, a higher SR is indicative of a better choice for a portfolio.

3.3.3. Maximum Drawdown (MDD)

The maximum drawdown (MDD) is the highest observed loss from a peak to a trough of a portfolio, before a new peak is attained. It is an indicator of downside risk over a specified time period. It is calculated as such:

$$MDD = -\frac{Through\ Value - Peak\ Value}{Peak\ Value} \tag{10}$$

The Maximum drawdown is oriented towards capital preservation, so a low MDD (in absolute value) is preferred as this indicates that losses of the portfolio or investment strategy will be smaller.

3.4. Portfolio Optimization

Portfolio optimization addresses the problem of allocate the inversion in the different assets following an objective function that should be optimized with respect to the portfolio vector.

Modern Portfolio Theory is based on the Markowitz [11] mean-variance framework which aim is to find a trade-off between the expected return and the risk of the portfolio measured by the variance.

The Mean-Variance Trade-Off Optimization can be formulated in different ways.

3.4.1. Risk Minimization Problem

Consisting in the portfolio variance minimization with the expected portfolio being above a given target μ_0 and the capital budget constraint $\mathbf{w}^T\mathbf{1} = 1$

$$\begin{aligned} &\underset{\mathbf{w}}{\text{minimize}}\ \mathbf{w}^{\mathrm{T}}\boldsymbol{\Sigma}\mathbf{w} \\ &\text{subject to: } \mathbf{w}^T\boldsymbol{\mu} \geq \mu_0 \\ &\mathbf{w}^T\mathbf{1} = 1 \end{aligned} \tag{11}$$

Minimum variance or volatility regardless the expected return can be formulated as:

$$\begin{aligned} &\underset{\mathbf{w}}{\text{minimize}}\ \mathbf{w}^{\mathrm{T}}\boldsymbol{\Sigma}\mathbf{w} \\ &\text{subject to: } \mathbf{w}^T\mathbf{1} = 1 \end{aligned} \tag{12}$$

which is a convex Quadratic Programming (QP) with only one linear equality constraint that yields the closed form solution:

$$\mathbf{w}_{MV} = \frac{\boldsymbol{\Sigma}^{-1}\mathbf{1}}{\mathbf{1}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{1}} \tag{13}$$

3.4.2. Return Maximization Problem

Alternatively, the expected return can be maximized with the variance under a given target σ_0^2 and the capital budget constraint $\mathbf{w}^T\mathbf{1} = 1$

$$\begin{aligned} &\underset{\mathbf{w}}{\text{maximize}}\ \mathbf{w}^T\boldsymbol{\mu} \\ &\text{subject to:} \mathbf{w}^{\mathrm{T}}\boldsymbol{\Sigma}\mathbf{w} \leq \sigma_0^2 \\ &\mathbf{w}^T\mathbf{1} = 1 \end{aligned} \tag{14}$$

Since the covariance matrix $\boldsymbol{\Sigma}$ is positive definite, this problem has a linear objective with linear and convex quadratic constraints, and thus can be efficiently solved.

3.4.3. Risk-Adjusted Return Maximization Problem.

The third option for Mean-Variance Trade-Off optimization problem formulation is to maximize a risk-adjusted return:

$$\underset{\boldsymbol{w}}{maximize}\ \boldsymbol{w}^T\boldsymbol{\mu} - \lambda\boldsymbol{w}^T\boldsymbol{\Sigma}\boldsymbol{w}$$

$$\text{subject to: } \boldsymbol{w}^T\mathbf{1} = 1 \tag{15}$$

where $\lambda \geq 0$ is a given trade-off parameter between the portfolio expected return and the variance. Again, this is a convex QP with only one linear constraint which admits a closed form solution as follows:

$$\mathbf{w}_{RA} = \frac{1}{2\lambda}\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{\mu} + \frac{2\lambda - \mathbf{1}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}}{\mathbf{1}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{1}}\mathbf{1}\right) \tag{16}$$

Note that the above problem formulations depend on the investor preferences by selecting the parameters μ_0 , σ_0^2 and λ. That is, the portfolios obtained setting the Mean-Variance Trade-Off in any of the alternative formulations are optimal depending on the investor's risk profile.

3.4.4. Sharpe Ratio Optimization

The Sharpe Ratio of a portfolio can be defined as the expected excess return with respect to the return of a risk-free asset r_f (for instance T-bills), normalized by the risk. Then, SR optimization problem can be formulated as follows:

$$\begin{aligned} &\underset{\mathbf{w}}{\text{maximize}}\ \frac{\mathbf{w}^T\boldsymbol{\mu} - r_f}{\sqrt{\mathbf{w}^{\mathrm{T}}\boldsymbol{\Sigma}\mathbf{w}}} \\ &\text{subject to: } \mathbf{w}^T\mathbf{1} = 1 \end{aligned} \tag{17}$$

The Sharpe ratio maximization problem is not a convex problem but can be reformulated in convex form as follows. First, the restriction $\mathbf{w}^T\mathbf{1} = 1$ implies that the numerator of the SR is equal to $\mathbf{w}^T(\boldsymbol{\mu} - r_f\mathbf{1})$. Furthermore, the constraint $\mathbf{w}^T\mathbf{1} = 1$ can be relaxed to $\mathbf{w}^T\mathbf{1} > 0$. Then, the SR maximization is equivalent to the minimization of the denominator with arbitrary constrained numerator $\mathbf{w}^T(\boldsymbol{\mu} - r_f\mathbf{1}) = 1$.

$$\begin{aligned} &\underset{\mathbf{w}}{\text{minimize}}\ \mathbf{w}^{\mathrm{T}}\boldsymbol{\Sigma}\mathbf{w} \\ &\text{subject to: } \mathbf{w}^T(\boldsymbol{\mu} - r_f\mathbf{1}) = 1 \\ &\mathbf{w}^T\mathbf{1} > 0 \end{aligned} \tag{18}$$

Any normalized solution of (18) so that the sum of the weights being equal to one is an optimal solution of (17).

The problem reformulated as:

$$\begin{aligned} &\underset{\mathbf{w}}{\text{minimize}}\ \mathbf{w}^{\mathrm{T}}\boldsymbol{\Sigma}\mathbf{w} \\ &\text{subject to: } \mathbf{w}^T(\boldsymbol{\mu} - r_f\mathbf{1}) = 1 \end{aligned} \tag{19}$$

is a convex Quadratic Problem with only one linear equality constraint that admits a closed form optimal solution:

$$\mathbf{w}_{SR} = \frac{1}{(\boldsymbol{\mu} - r_f\mathbf{1})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - r_f\mathbf{1})}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - r_f\mathbf{1}) \tag{20}$$

Assuming, as observed in practice [9], that $\mathbf{w}_{SR}^T\mathbf{1} > 0$, we can conclude that $\mathbf{w}_{SR}$ is also the optimal solution of (18).

4. Deep Reinforcement Learning

Reinforcement learning is considered to be a subfield of machine learning. At its most basic analogy it consists on training a program the way we, humans, learn in real life, that is by reward if we do something good or by punishment if we do something wrong.

In this chapter the basic DRL concepts are introduced [12]. Moreover, we review the major components of a reinforcement learning algorithms, and finally we will dive deep into the specific algorithms that will be used in the project.

4.1. Dynamical Systems

Reinforcement learning is suitable for optimally controlling dynamical systems which change over time.

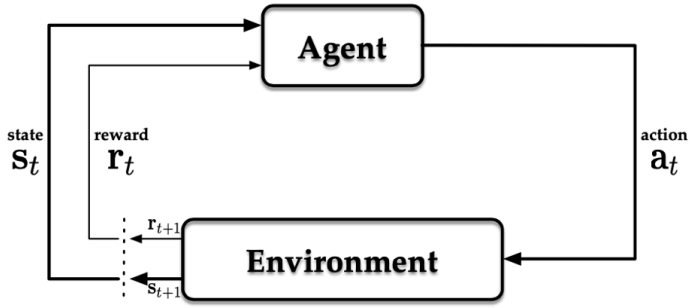


Figure 2. Dynamical system in Reinforcement Learning [12]

A controller (agent) receives the controlled state of the system (environment) and a reward associated with the last state transition. Then, it calculates a control signal (action) which is sent back to the system.

In response, the system makes a transition to a *new state* and the cycle is repeated iteratively. The goal of reinforcement learning is to train the agent into successfully interacting with the environment and learning a way of controlling the system (policy) which maximizes the total reward function over time.

4.1.1. Action

The Action $a_t \in A$ is the signal which works as a control and the agent sends to the environment at time *t*. Therefore, it is the way the agent interacts with the system. These interactions lead to the modification of the reward signal.

The action space A refers to the defined set of actions the agent is allowed to take.

4.1.2. Reward

The reward: r_t is a scalar feedback function that indicates how well the agent is behaving at discrete given time t.

The agent through his actions, works to maximize the total cumulative reward, over a sequence of discrete steps.

Reinforcement learning addresses sequential decision-making tasks. In other words, by training agents that optimize delayed rewards and evaluating the long-term consequences of its actions, it is able to sacrifice immediate reward to gain more long-term reward.

This property of DRL agents makes them a very attractive option for financial applications, such as investments, where time horizons range from days to years or even decades.

It is fundamental to choose the right reward function for each application when aiming to get the best performance of an algorithm.

4.1.3. State and Observation

The state, $\boldsymbol{s}_t \in \boldsymbol{S}$, usually refers to the environment state and the agent state.

The environment state $\boldsymbol{s}_t^e$ is the internal representation of the system, used in order to determine the next observation o_{t+1} and reward r_{t+1}. The environment state is usually invisible to the agent.

The history $\boldsymbol{h}_t$ at time t is the sequence of observations, actions and rewards up to time step t, such that:

$$\mathbf{h}_t = (o_1, a_1, r_1 \ldots o_t, a_t, r_t) \tag{21}$$

with o_t being the observation at time t.

The agent state $\boldsymbol{s}_t^a$ is the internal representation of the agent about the environment. It is used in order to select the next action a_{t+1} and it can be any function of the history:

$$\boldsymbol{s}_t^a = f(\boldsymbol{h}_t) \tag{22}$$

The term state space S defines the possible states the agents can observe. It can be:

4.2. Components of DRL

4.2.1. Return

The return is the total accumulation of rewards attained by the system. Let γ be the discount factor of future rewards. The future discounted reward is given by

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \qquad \gamma \in [0,1] \tag{23}$$

4.2.2. Policy

The Policy, π ,is the function which determines the agent's behavior:

$\pi: S \rightarrow A$ where S and A are respectively the state space and the action space.

4.2.3. Value Function

State-value function, $V_\pi(s)$ represents how good the state an agent is in is. It is the expected return, G_t, starting from state $\boldsymbol{s}$, which follows a policy π [13] that is:

$$V_\pi(s) = \mathbb{E}\left[\sum_{i=1}^{T} \gamma^{i-1} r_i\right] \qquad s \in S \tag{24}$$

The optimum value function is the one that has the higher value for all states. And the optimal policy is the one which maximizes the optimal Value function.

State-Action function, Q, or simply put the Q function. is the expected return, G_t, received by the agent starting from state s, upon taking action a:

The optimal Q function is the one with maximum expected reward.

In DRL these value functions and the policy are usually approximated using Deep Neural Networks.

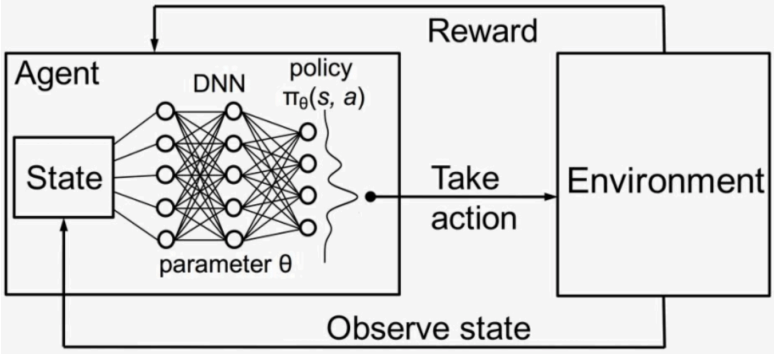


Figure 3. Dynamical system in Deep Reinforcement Learning Model

Refers to the agent's representation of the environment. A model predicts the next state, $\boldsymbol{s}_{t+1}$ of the environment, and the corresponding reward, $\boldsymbol{r}_{t+1}$, given the current state, $\boldsymbol{s}_t$, and the action taken, $\boldsymbol{a}_t$, at time step, we can classify DRL algorithms in three different categories [14]:

4.2.4. Critic-only approach

A learned value function allows the agent to compare different actions. In the decision-making process, the agent observes the current state of the environment and chooses the action that provides the best outcome.

The advantage of this approach is the high flexibility of the reward function and its applicability to a wide variety of problems. The reward function does not need to be differentiable in this case. Also, a discount factor is used, this allows to control the tradeoff between immediate and future reward.

This approach is the most popular approach in financial markets.

4.2.5. Actor-only approach

In this case the policy is learned instead of approximating the value function. The main advantages that It provides a continuous action space which allows the agent to carefully interact with the environment, and usually converges faster. The most noticeable disadvantage of the actor-only approach is the need for a differentiable reward function, limiting the reward schemes that can be modeled.

4.2.6. Actor-critic approach

Aims at combining the advantages of the other two approaches. It is comprised by two agents, the actor and the critic.

The actor determines the actions and forms the policy of the system. The critic evaluates these actions and computes the discounted future reward as output.

The idea is to gradually adjust the policy parameters of the actor in a way that it maximizes the reward predicted by the critic.

We have found very few studies employing actor-critic RL in financial markets.

This project will focus on the Actor to critic approach fundamentally using three different algorithms. In this section we explain each one of them.

4.3. Algorithms

4.3.1. Advantage Actor Critic (A2C)

A2C improves the policy gradient updates. It uses an advantage function $A(s_t, a_t)$ in order to reduce the variance of the policy gradient. Instead of only estimating the value function, the critic network estimates the advantage function. So, the evaluation of an action depends on how good the action is, but also how much better it can be. This makes the model more robust.

A2C uses copies of the same agent to update gradients with different data samples. Each agent works independently and interacts with the environment. In each iteration, after all agents have finished calculating their gradients, A2C what's called a coordinator to pass the average gradients over all the agents to a global network. So that the global network can update the actor and the critic network. The presence of a global network increases the diversity of training data. The synchronized gradient update is more cost-effective, faster and works better with large batch sizes.

The objective function for A2C is:

$$\nabla J_\theta(\theta) = E[\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) A(s_t|a_t)] \tag{25}$$

where $\pi_\theta(a_t|s_t)$ is the policy network, $A(s_t|a_t)$ is the Advantage function and can be written as:

$$A(s_t|a_t) = Q(s_t|a_t) - V(s_t) \text{ or,}$$
$$A(s_t|a_t) = r(s_t, a_t, s_{t+1}) - \gamma V(s_{t+1}) - V(S_t) \tag{26}$$

4.3.2. Deep Deterministic Policy Gradient (DDPG)

DDPG combines the frameworks of Q-Learning and Policy Gradient and uses four different neural networks as a way to approximate function. It learns directly from the empirical observations through policy gradient.

At each timestep the DDPG agent performs an action a_t at s_t, then it is awarded with a reward r_t and finally arrives at the next state s_{t+1}. These transitions (s_t, a_t, s_{t+1}, r_t) are stored in the replay buffer R. To update the Q-value a batch of N transitions is taken from R and it is computed such that:

$$y_i = r_i + \gamma Q'\left(s_{i+1}, \mu'\left(s_{i+1}|\theta^{\mu'}, \theta^{Q'}\right)\right) \text{ for } i = 1 \dots N \tag{27}$$

Lastly, the critic network is then updated by minimizing the loss function $L(\theta^Q)$.

4.3.3. Proximal Policy Optimization (PPO)

In many Policy gradient methods, the policy update is unstable. PPO tries to simplify the objective of Trust Region Policy Optimization by introducing a clipping term to the objective

function. It is used to control the policy gradient update and make sure that the new policy will not be too different from the previous one.

Let's assume the probability ratio between old and new policies is expressed as:

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta old}(a_t|s_t)} \tag{28}$$

Them, the clipped objective function of PPO is:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[\min\left(r_t(\theta)\hat{A}(s_t,a_t), clip(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}(s_t,a_t)\right)\right] \tag{29}$$

where $rt\ (\theta)A(st,at)$ is the normal policy gradient objective and $\hat{A}(s_t,a_t)$ is the estimated advantage function.

In essence PPO discourages large policy change move outside the clipped zone. Therefore, it improves the stability of the policy training networks by restricting the policy update at each training step.

5. Methodology / project development:

Our project has been developed working around a previous project developed in [15] and improving its methodology.

To do so a first phase of learning and researching about Financial Signal Processing and Deep Reinforcement Learning was fundamental. Afterwards we had to comprehend in its entirety the previous project to be able to propose some innovations which could lead to a possible improvement on performance. Finally, the proposals were developed, implemented, and tested over the original strategy.

In this section we firstly aim to briefly explain the basis of the mentioned article and its developed strategy, so it serves as a summary of the article itself for the reader who might be curious about our project. And finally, we explain the improvements that have been done to the Ensemble Strategy in order to improve it.

5.1. Original Ensemble Strategy (OES)

5.1.1. Introduction

In this paper, an ensemble strategy is proposed which combines three Actor - Critic deep reinforcement learning algorithms (presented in section 3.3) to find the optimal trading strategy. The idea is that by applying the ensemble strategy, the trading strategy becomes more robust in a highly dynamic environment.

The algorithm works in iterations as such:

1. **Training** of all the algorithms (training window starts at 02/01/2009 and ends in 10/01/2015)
2. **Validation** of all the algorithms. SR to measure validation performance is computed for each one (validation window starts at 11/01/2015 and is 63 days long)
3. **Trading** is done using the algorithm with higher SR (trading window starts right after of validation and lasts for 63 days)

4. The training window for next iteration now becomes larger because it includes the validation period on this iteration. While the validation and trading window don't change, just shift 63 days in time.

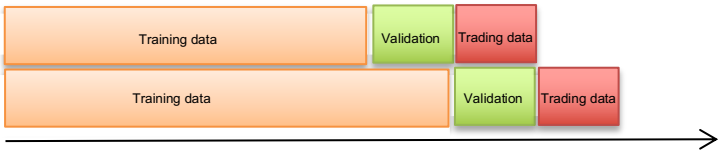


Figure 4. Training, validation and trading windows.

5.1.2. Problem description

The system can be described by the following figure

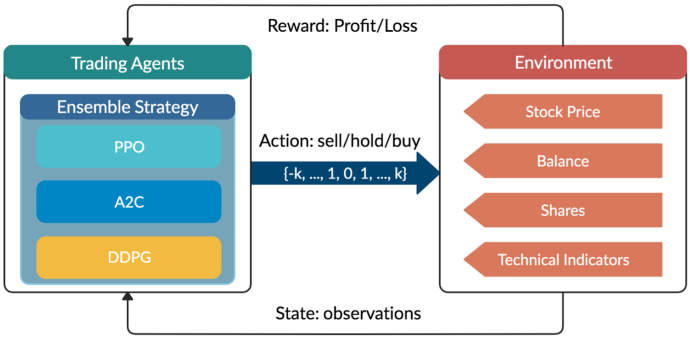


Figure 5. Original Ensemble strategy [15]

5.1.2.1. Trading Model for one stock

- State $\boldsymbol{s} = [\boldsymbol{p}, \boldsymbol{h}, b]$ where $\boldsymbol{p}$ represents the stock prices, $\boldsymbol{h}$ represents the stock shares, and $\boldsymbol{b}$ represents the remaining balance
- Action $\boldsymbol{a}$: represents the vector of possible actions, Buying, selling or holding. Interacts with $\boldsymbol{h}$
- Reward $r(s, a, s')$: reward of taking action a in state s and going to state s'.
- Policy $\pi(s)$: the trading strategy at state s.(i.e., the probability distribution of actions at said state)
- Q-Value $Q_\pi(s, a)$: expected reward of taking action a in state s following policy π

5.1.2.2. Trading constraints

Some constraints are considered

- Market Liquidity: It is assumed that our actions don't have an impact on the market conditions thus the orders are executed immediately at the close price.
- Non-negative balance: the actions cannot lead to $b < 0$
- Transaction cost: transaction cost is assumed to be 0,1 % of each trade.
- Risk aversion for market crash: the turbulence index is introduced [16]. When turbulence index is high it indicates sudden sell offs that may yield a stock crash. If turbulence index surpasses the set threshold all shares are sold and trading is halted.

$$turbulence = (y_t - \mu)^T \Sigma^{-1} (y_t - \mu) \quad (30)$$

5.1.2.3. Return

The reward is defined as the change of portfolio value when action a is taken at state s and arriving at state s'. So, the goal is to maximize portfolio returns.

5.1.3. Stock market environment

Before training the agents, the trading environment is built to simulate real world situations. Multiple information is added to this environment such as: stock prices, stock holdings and technical indicators.

5.1.3.1. Environment for Multiple stocks

The State Space used is a 181-dimensional vector such as: $[b_t, \boldsymbol{p}_t, \boldsymbol{h}_t, \boldsymbol{M}_t, \boldsymbol{R}_t, \boldsymbol{C}_t, \boldsymbol{X}_t]$ with

- b_t: available balance at time step t.
- $\boldsymbol{p}_t$: adjusted close price for each stock
- $\boldsymbol{h}_t$: shares owned of each stock
- $\boldsymbol{M}_t$: Moving Average Convergence Divergence (MACD): momentum indicator.
- $\boldsymbol{R}_t$: Relative Index Strength (RSI): Indicates whether a stock is oversold or overbought.
- $\boldsymbol{C}_t$: Commodity Channel Index (CCI): Compares the current price to the average price to indicate buying or selling.
- $\boldsymbol{X}_t$: Average Directional Index (ADX): Identifies trend strength.

The Action Space for each stock is define such as: $\{-k, \dots, -1, 0, 1, \dots k\}$: where k and $-k$ represent the number of shares we can buy or sell.

5.1.3.2. Ensemble strategy

The strategy selects the best performing agent to trade based on the Sharpe Ratio obtained in the validation period. The idea behind this is that each algorithm is sensitive to different types of trends (i.e., bullish or bearish market conditions).

6. Innovation

The goal of this project relies on improving the performance of the strategy proposed on the previous section as a whole, be it by lowering risk or improving return. To do so, some innovations had to be thought about, modelled, implemented and finally tested.

The goal of this project relies on improving the performance of the trading agent as a whole, be it by lowering risk or improving return. To do so, some innovations had to be thought about, proposed, modelled, implemented and finally tested.

During the course of the project many ideas were proposed as possible implementations to improve said performance. However, not all of them were possible to model nor to apply or test, due to a lack of resources, time or knowledge.

In this chapter the innovations which have been finally implemented to the Original Ensemble Trading Strategy are explained.

6.1. Differential Sharpe ratio as reward function

As it has been explained before, the OES works by implementing at each Trading window one of the three DRL Algorithms, A2C, PPO or DDPG.

In the original article the reward function $r(s, a, s')$ happens to be the return of a trading period:

$$r = account\_value(T_N) - account\_value(T_0) \tag{31}$$

This means the agent works in order to maximize the profit accumulated over a trading period. This approach is very simple computationally speaking and very good results have been obtained. While it may seem to be a good enough strategy, using the return as a reward function comes with two implicit problems.

Return is a very important parameter to take into consideration and try to maximize it, but so is risk. In the original approach no risk related parameters are factored in the reward function. This can cause the agent to take riskier positions and thus create more downside opportunities for our portfolio.

As it has been previously explained the strategy uses the Sharpe Ratio obtained by each algorithm during the validation period in order to choose the agent that will be used for the trading. While this may not be a problem, it made us ask ourselves whether making use of the Sharpe Ratio as the reward function would in fact provide better results than previously or not.

We have previously presented the Sharpe ratio as a very efficient metric to measure an investment strategy performance. Because it considers both return and risk.

$$SR_{1:T} = \sqrt{T}\ \frac{\mathbb{E}[\boldsymbol{r}_{1:T}]}{\sqrt{Var[\boldsymbol{r}_{1:T}]}} \tag{32}$$

With T being the number of samples considered in the calculation of the empirical mean and the standard deviation. That brings the problem of using empirical estimation for the calculation of both parameters, which makes the SR not an appropriate choice for online learning, such as our case

However, we can implement the Differential Sharpe Ratio (DSR) [17] as a valid reward function, because it allows online learning.

The DSR is advantageous to use on-line performance functions both to speed the convergence of the learning process (since parameter updates can be done during each forward pass through the training data), and to adapt to changing market conditions during live trading.

The DSR is obtained by expanding SR_t to the first order in the decay parameter (η).

$$SR_t \approx SR_{t-1} + \eta \frac{dSR_t}{d\eta}|_{\eta=0} + O(\eta^2) \tag{33}$$

The differential Sharpe Ratio, $\boldsymbol{D_t}$, is defined as:

$$D_t \triangleq \frac{dSR_t}{d\eta} = \frac{B_{t-1}\Delta A_t + \frac{1}{2}A_{t-1}\Delta B_t}{(B_{t-1} - {A_{t-1}}^2)^{\frac{3}{2}}} \tag{34}$$

where A_t and B_t are the exponential moving estimates of the first and second moments of r_t , given by:

$$A_t = A_{t-1} + \eta \Delta A_t = A_{t-1} + \eta(r_t - A_{t-1}) \tag{35}$$

$$B_t = B_{t-1} + \eta \Delta B_t = B_{t-1} + \eta(r_t{}^2 - B_{t-1}) \quad (36)$$

Using the differential Sharpe Ratio for reward, results in the multistep maximization of the SR, which balances risk and profit, and hence it is expected to lead to better strategies, compared to cumulative log or simple returns.

In order to optimize the DSR It is demonstrated in [17] the importance for the η parameter to be the inverse of the temporal window used for the parameter actualization. This presents a tradeoff between the adaptability to variations in the scenario and the quality of the estimation.

6.2. Data Expansion

As it has been explained previously the OES is feeded with a dataset conformed by the financial data referred to the 30 stocks of the Dow Jones industrial Index.

Now one may ask for the reason to limit the investment strategy to a total universe of 30 unique stocks.

Theoretically, if we were to have a wider universe of stocks to choose from wiser decisions could be taken without limiting ourselves to the limiting factor that supposes to have a small number of possible stocks to buy.

One of the motives behind the previous reasoning is that The Dow Jones Industrial Average is an index comprised by the 30 biggest corporations in the US. However There exist around 630.000 companies that are listed and publicly traded in different stock exchanges worldwide. And although the DJIA might be a very strong indicator of the US economy, even the world's economy, it doesn't necessarily mean that those 30 companies are the biggest gainers daily (a stock is called a gainer when it grows a certain percentage, from one day to another. This percentage is much higher than what the average stock may return daily).

We would like to take advantage of the gainers in the market and also avoid the higher losers. And to be exposed to these outliers we need to have access to more stocks.

Also, we would like to take advantage of having exposure to all the different existent sectors to hedge our portfolio against risk and from a possible adverse market situation.

Finally, it is known that certain industries may present cyclical patterns which our agents could learn and exploit (i.e., energy sector the usually presents growth during winter months whereas it contracts during summer months). And by only considering 30 stocks we might not have the opportunity to be present in all the sectors of the economy, which are:

- Financial
- Communication Services
- Real Estate
- Technology
- Industrial
- Healthcare
- Consumer/retail
- Energy
- Utilities
- Basic materials

We believe that by enlarging the universe of stocks which the agents have exposure to and can choose from will improve performance of the OES. We expect there to be a tradeoff between execution time and performance.

The solution proposed relies on using the 500 companies listed in the S&P500, one of the most known indices in financial markets which is considered to be the benchmark for all financial investment strategies or simply the "market" as such.

This index contains the 500 most important companies of the United States and it comprises companies from all the sectors in the economy.

With the aim of comparing both the original strategy with the proposed one we will be using the same start and end dates.

The first step is to download all the necessary data. In this case we don't have access to the database used in the original article. A data scrapping script has been developed (getdata.py) which downloads the financial data (Open price, close price, high, low, adjusted and volume) from the Yahoo finance database of all the stocks comprised in the S&P500 in the specified timeframe.

However, two main problems arise. The first one being that the stocks in the S&P500 are dynamic, meaning that they change overtime. Given that all companies experience some growth (positive or negative) means that not all the companies that were in the S&P500 10 years ago are necessarily in it today, or vice versa. Some might even have gone out of business, or some might even be younger than that.

Secondly, the format in which the data is downloaded is very different than the one needed by the program, the structure is different and what is more, we need to add in the financial indicators and the turbulence index for the new data.

The solution to both problems explained above is the same. Data manipulation and preprocessing needs to be done. The first issue is addressed by only considering the companies which have data for the whole-time interval specified. This specifically turned out to be 427 out of 500.

The second issue has been solved by using the class preprocessors.py and by manipulating data frames, until the right structure has been achieved.

Also, some minor changes to the code need to be done to consider the right amount of unique stocks and the correct sizing of state vector.

6.3. Combining DRL strategy

The second proposition that came up was to linearly combine the three algorithms used (A2C, PPO and DDPG) instead of choosing one of them only in each iteration. The main reason behind this reasoning was the idea that each algorithm performs best under certain situations:

A2C: Good performance when facing a bearish market.

PPO: Best performance when facing a bullish market.

DDPG: Better performance when facing a bullish market, though not as good as PPO.

We have developed a combining DRL trading that consist of distributing the total balance between the three different strategies instead of selecting one of them. The proposed criterion for allocation optimization has been the Sharpe Ratio maximization.

The combining DRL consist in finding the vector of weights $\mathbf{w}_{SR} \in \mathbb{R}^3$ that will define the proportion of the total balance that should be assigned to each DRL strategy following the SR optimization problem (18):

$$\begin{aligned} &\underset{\mathbf{w}}{\text{minimize}}\ \mathbf{w}^{\mathrm{T}}\boldsymbol{\Sigma}\mathbf{w} \\ &\text{subject to: } \mathbf{w}^T\left(\boldsymbol{\mu} - r_f\mathbf{1}\right) = 1 \\ &\mathbf{w}^T\mathbf{1} > 0 \end{aligned} \tag{37}$$

The expected return $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$ of the portfolio returns obtained with the different strategies can be estimated by sample averages along the validation interval T.

$$\hat{\boldsymbol{\mu}} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{r}_t \qquad \boldsymbol{\mu} \in \mathbb{R}^3 \tag{38}$$

$$\boldsymbol{\Sigma} = \frac{1}{T}\sum_{t=1}^{T}(\mathbf{r}_t - \hat{\boldsymbol{\mu}})(\mathbf{r}_t - \hat{\boldsymbol{\mu}})^T \qquad \boldsymbol{\Sigma} \in \mathbb{R}^{3\times 3} \tag{39}$$

In our problem, we found that $\mathbf{w}_{SR}$ does not always verify the constraint $\mathbf{w}_{SR}^T\mathbf{1} > 0$. Then we cannot consider the closed form (20) and we apply second order cone programming to obtain the optimal solution of problem (18)

7. Results

In this section we present the results obtained from the innovations proposed for the project. All simulation were made with an initial account value of 1.000.000 €.

7.1. Differential Sharpe Ratio as a Reward Function

We were able to implement the theoretical proposition of the DSR as a reward function successfully. We also computed the results we would get for the original ensemble strategy using the return as the reward function, all the DRL algorithms used in the ensemble strategy but separately and finally the DJI index to be used as a benchmark.

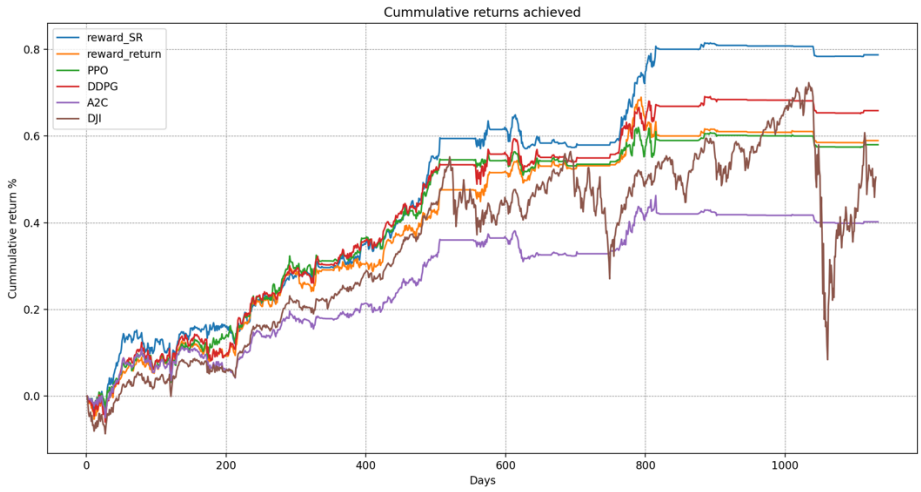


Figure 6. DSR CR compared to OES, A2C, DDPG, PPO, and DJI index

Figure 6 shows the accumulated returns resulting from the trading period. As it is shown, the results obtained by using the DSR as a reward function outperform the original ensemble strategy as well as the three actor-critic based algorithms. This result becomes very noticeable at 800 days. Also, we would like to point out the usefulness of using the turbulence as a factor to consider in the trading. Note that in Figure 6 around index = 1050 the DJI price drops dramatically, due to the COVID-19 sell-off. If we had bought the market and were using the Buy and Hold strategy, our portfolio would have experienced a 37% drop in value. However, note that by making use of DRL algorithms and the turbulence indicator during the COVID-19 Sell off, positions are sold and buying is withheld, therefore, returns stay almost static, and capital is preserved.

Figure 7 shows the MDD. At first sight it is very appreciable that all the algorithms using DRL perform much better in terms of MDD than the benchmark which would be, buying the market.

To sum up the results obtained by changing the reward function we have computed Table 1, in which we show the three-performance metrics presented in section (3.3). As it is observable, The Sharpe_reward strategy developed by us is the one which shows better performance overall. Despite the MDD not being the lowest in the list (-8,7% against -6,2% as the minimum) it is very far from the MDD the benchmark suffers from at -37%. What is more, CR are much larger than the ones obtained by other implementations, and the SR, is the best of the list which means, our strategy has the better return-risk ratio, outperforming the Return_reward and so, validating our first hypothesis.

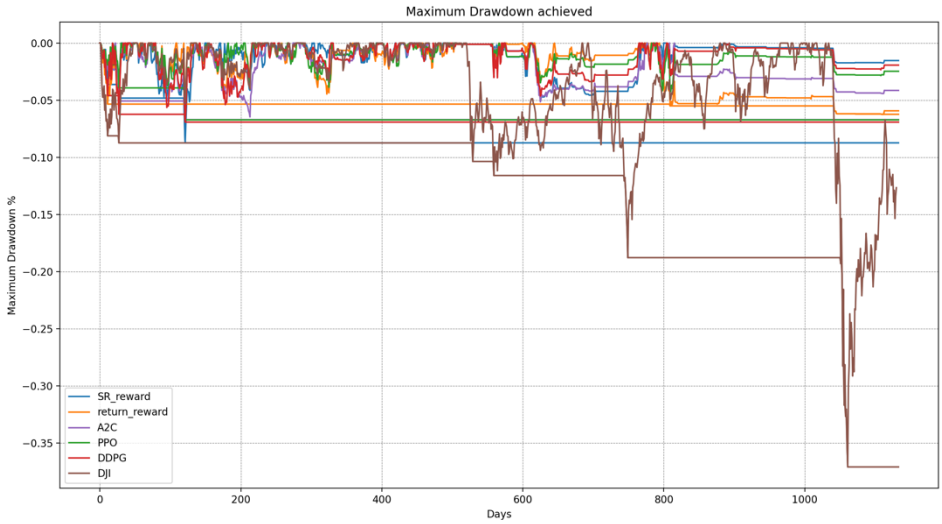


Figure 7. DSR MDD to OES, A2C, DDPG, PPO, and DJI index

Table 1. DSR performance evaluation comparison

	Sharpe_reward	**Return_reward**	**A2C**	**PPO**	**DDPG**	**DJI**
CR	78,74	58,96	40,23	58,01	65,87	50,52
SR	1.6	1.36	1.12	1.32	1.19	0.54
MDD	-8,7%	-6,2%	-6,7%	-6,7%	-6,9%	-37%

7.2. Data Expansion

In this second hypothesis we believed that by enlarging the dataset which we were working with and providing a considerably bigger universe of stocks would provide more options to the algorithm and therefore provide better performance overall.

Firstly, as explained in section () we did not have access to the dataset used by the original authors. So, getting the data in the same format as theirs was a struggle.

A script was developed to scrap the data from yahoo finance and pre-process the data and its structure. After doing so we were left with a dataset comprised by 441 stocks instead of 500 that we had initially for the reasons explained in section (). However, the dataset was still big enough to prove our point.

After that we had to compute the financial indicators which had been used in the original project (MACD, RSI, CCI and ADX) for each stock.

We faced the problem of not being able to compute the turbulence index for the data in the S&P500 Database.

The results on this section are presented without using the turbulence index, however the original ensemble strategy, using the dataset comprised by the S&P 500 stocks and the dataset with DJI-30 stocks, is still compared with the DJI benchmark.

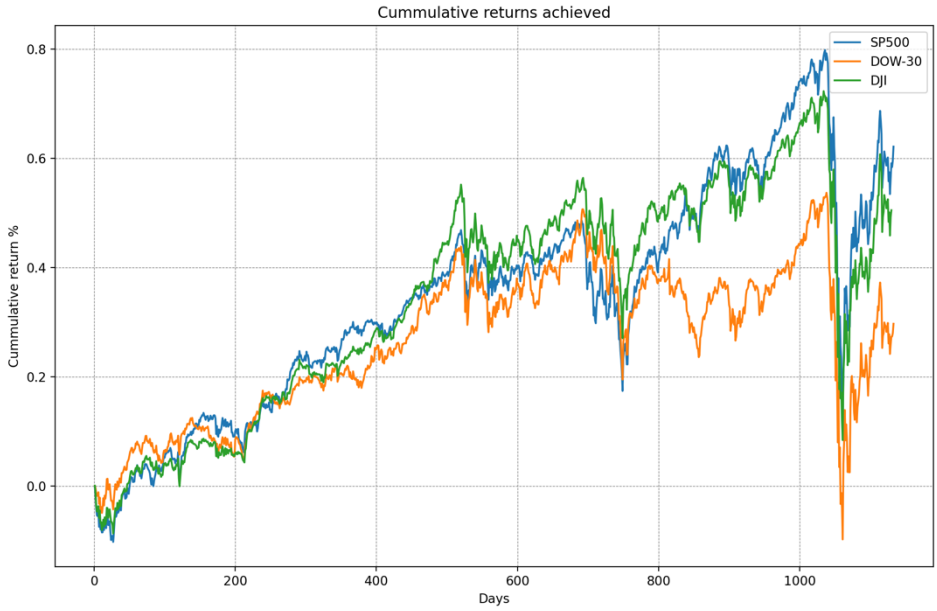


Figure 8. S&P500 CR compared with DOW-30 and DJI index

As it was expected, from figures 8 and 9, it can be noticed that using a larger dataset with more stocks to choose from improves the Cumulative return obtained by the Ensemble Strategy.

However, the main downside of this approach is the impact expanding the dataset has had on execution times when running the program. The average execution time for the original program were 120 minutes whilst when expanding the dataset was 470 minutes.

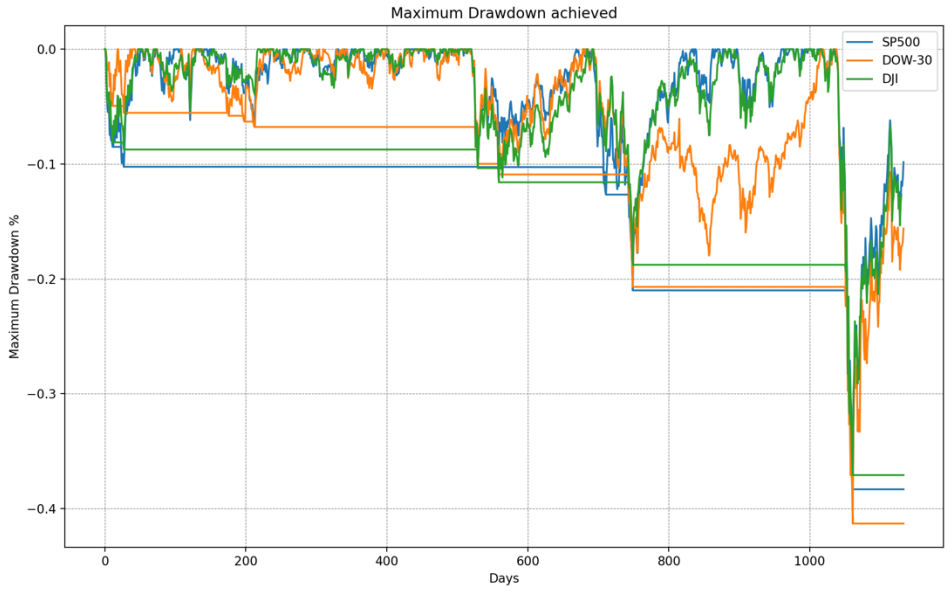


Figure 9. S&P500 MDD compared with DOW-30 and DJI index

Table 2. S&P500 performance evaluation comparison

	S&P500	**DJ-30**	**DJI**
CR	62,17	29,7	50,5
SR	0,63	0,36	0,55
MDD	-38,32	-41,31	-37,09

The Interpretation of the results in Table 2 is that our approach has significantly improved all of the Performance metrics over the original approach of just using 30 stocks, making it a better approach for future implementation.

Also, we can appreciate the impact of the turbulence index on this strategy is very significant. Despite the Original strategy in the previous section being well above on performance compared to the DJI benchmark, it shows now a lower performance. Our strategy is closer to the benchmark, however using the it still performs better.

It would be interesting to compute the turbulence index for out dataset and compare the results altogether. Results should be significantly better than the ones obtained, judging by the drop on performance the original strategy has suffered in this case.

7.3. Combining DRL strategy

It has been difficult to implement this strategy. The number of variables that had been taken into account are several which increase the difficulty of it.

First of all, the solution proposed was not enough. We observed that some coefficients were negative. The solution was to implement a second order cone solver which optimized the coefficients to be positive. The one used was: cvxopte_solve_qp(P,q,G,h,A,b)

Sometimes we realized that to complement a negative coefficient the solver would assign a very small value that multiplied by the account balance would not make sense (i.e., the new amount would be less than a cent, which in financial markets doesn't make sense).

We have had to add a number of constraints like these one and also, the code had to change a lot due to the algorithms interacting with each other between intervals.

We thought that adding all the limiting factors would make the strategy decrease in performance. However, by looking at Figure 10 and Figure 11 we see that our strategy is the one with more cumulative return overall. The maximum Drawdown is also the lowest.

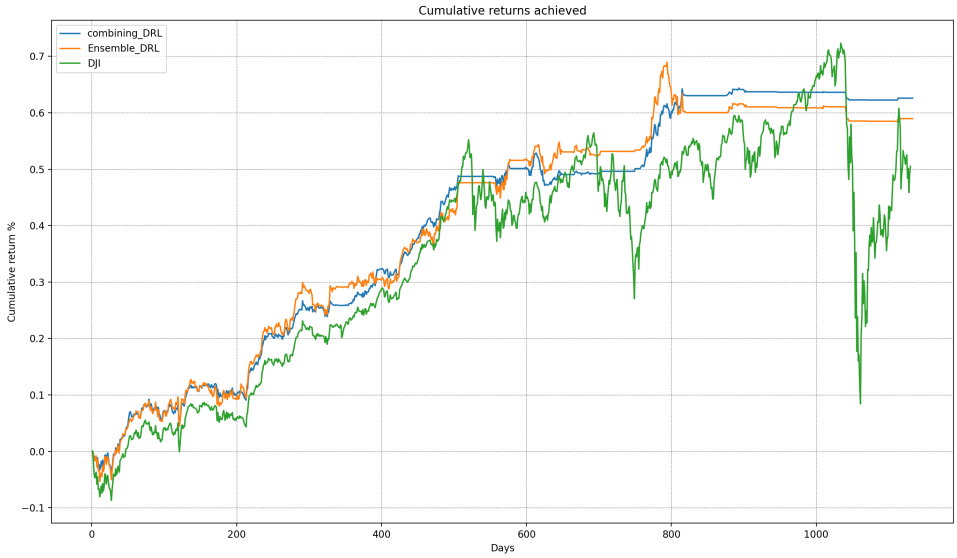


Figure 10. Combining DRL CR compared with Ensemble DRL and DJI index

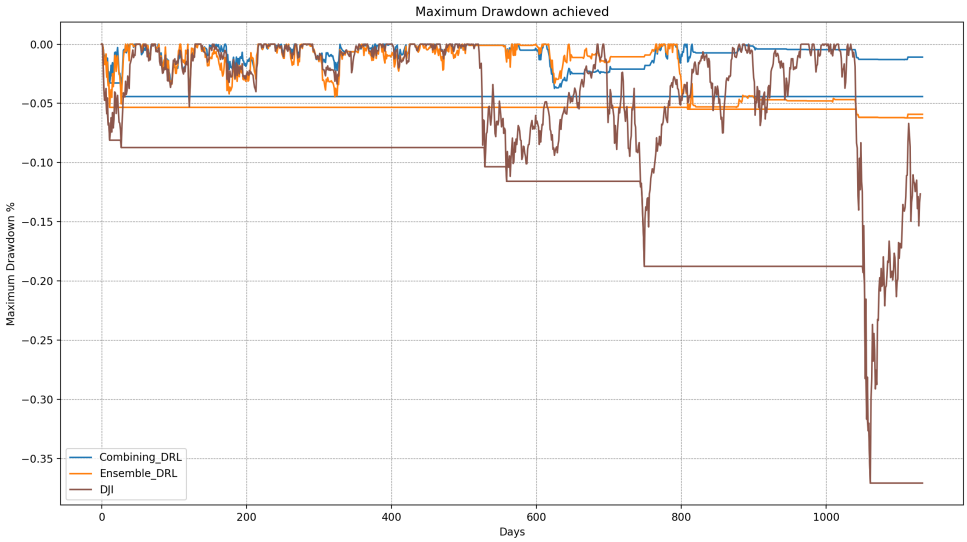


Figure 11. Combining DRL MDD to Ensemble DRL, and DJI index

By looking at the results obtained in table 3 we can see that not only the Combining strategy was the best in terms of return, but also, and most importantly it over-exceeded the Sharpe Ratio expectations compared to the Ensemble DRL and, obviously the benchmark. That means our strategy is far less risky than the Ensemble DRL.

We attribute this improvement to the fact that when the economic scenario is uncertain and there are no stablished trends, the algorithm takes advantage of combining the different strategies (which perform best in different situations), when only using one could simply mean choosing the wrong approach and having losses. We believe that by combining we don't necessarily improve returns in a drastic manner but yet we manage risk far better.

Table 3. Combining DRL performance evaluation comparison

	Combining DRL	**Ensemble DRL**	**DJI**
CR	62,58	58,96	50,5
SR	1.88	1.36	0,55
MDD	-4,42%	-6,2%	-37,09

8. Budget

The budget for this project is:

- Computer: An iMac 27" with an original cost of 2099,00€ will be used for the realization of the project.
- Salary: A Telecommunications student is paid 9,00 € per hour. The total working hours will be: 450€
- Software needed: A 3-month subscription to PyCharm IDE for professionals will be needed: 19,90/ month.

	CONCEPT	AMOUNT
ITEM1	iMac 27"	2099,00€
ITEM 2	Student Salary	4050,00€
ITEM 3	PyCharm IDE	59,70€
TOTAL		**6208,70€**

The Total budget for the project is: **6208,70€**

9. Conclusions and future development:

Our Innovations turned out to be effective for our purpose. In each case we improved the performance of the original Ensemble algorithm and so, achieved our goal. Considering the Differential Sharpe ratio as a reward function outperformed all the other innovations by far, providing a better result than expected. Enlarging the dataset also provided better performance, though worsening execution times a lot. And lastly the Combining DRL Strategy proved to be the best of all approaches in terms of Sharpe Ratio.

As of future developments we are interested in exploring other DRL models. Moreover, as we have seen, the turbulence index has shown a very prominent effect on the results of the project. It would be very interesting to perform innovation 2 including the index. Being financial risk indicator very fundamental for our strategy it would be interesting to study the effect of other indicators in our result such as the Systematic Risk indicator [18].

Also, one of the possible improvements to consider would be to factor other data in our model such as financial fundamentals, other technical indicators, or alternative data.

Improve the realism of market conditions simulated, such as the order execution time and also the cost per trade, though access to better databases would be needed.

As for the coding aspects two improvements might be needed. Firstly, explore computational solutions for improving the execution time and upgrading to python 3.9 where TensorFlow libraries are updated.

Bibliography:

[1] R. Neuneier, “Enhancing Q-learning for optimal asset allocation”. Conference on Neural Information Processing Systems (NeurIPS), 1997.

[2] F. Bertoluzzo and M. Corazza, “Testing different reinforcement learning configurations for financial trading: introduction and applications”, Procedia Economics and Finance 3 68–77, 2012.

[3] G. S. Atsalakis, K. P. Valavanis, “Surveying stock market forecasting tech- niques–part II: Soft computing methods”, Expert Systems with Applications, 36 (3), 5932–5941, 2009.

[4] Y. Fang, X. Liu, and H. Yang, “Practical machine learning approach to capture the scholar data driven alpha in AI industry”, In 2019 IEEE International Conference on Big Data (Big Data) Special Session on Intelligent Data Mining. 2230–2239, 2019.

[5] H. Yang, X. Liu, and Qi. Wu, “A practical machine learning approach for dynamic stock recommendation”. In IEEE TrustCom/BiDataSE, 1693–1697, 2018

[6] Z. Jiang, D. Xu and J. Liang, “A deep reinforcement learning framework for the financial portfolio management problema”. arXiv: 1706.10059, 2017.

[7] F. Soleymani, E. Paquet, “Financial portfolio optimization with online deep reinforcement learning and restricted stacked autoencoder—DeepBreath”, Expert Systems with Applications, Volume 156, 2020.

[8] A. Filos, “Reinforcement Learning for Portfolio Management”, M.S. thesis, Imperial College of London, London, UK, 2018.

[9] X. Zhang and F. Wang, “Signal Processing for Finance, Economics and Marketing”, IEEE Signal Processing Magazine, May 2017.

[10] Y. Feng and D. P. Palomar. “A Signal Processing Perspective on Financial Engineering”. Foundations and Trends in Signal Processing, vol. 9, no. 1-2, pp. 1–231, 2015.

[11] H. Markowitz. 1952. Portfolio selection. Journal of Finance 7, 1 (1952), 77–91.

[12] R. Sutton and A. Bario, “Reinforcement Learning: An Introduction”, The MIT Press Cambridge, Massachusetts London, England, 2017.

[13] Szepesvári, Csaba “Algorithms for reinforcement learning”. Synthesis Lectures on Artificial Intelligence and Machine Learning 4.1, pp. 1–103, 2010.

[14] T. Fischer, "Reinforcement learning in financial markets - a survey", FAU Discussion Papers in Economics 12/2018, Friedrich-Alexander University Erlangen-Nuremberg, Institute for Economics.

[15] H. Yang et al, “Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy”, 2020 ACM International Conference on AI in Finance ICAIF ’20, October 15–16, 2020, New York, NY, USA.

[16] M. Kritzman, CFA and Y. Li, “Skulls, Financial Turbulence and Risk Management”, Financial Analysts Journal, Volume 66 • Number 5, 2010.

[17] J. Moody et al, “Performance Functions and Reinforcement Learning for Trading Systems and Portfolios”, Journal of Forecasting, Volume 17, Pages 441-470, 1998.

[18] M. Kritzman, Y. Li, S. Page, and R. Rigobon, “Principal Components as a Measure of Systemic Risk”, MIT Sloan School Working Paper 4785-10

Glossary

ADX	Average Directional Index
A2C	Advantage Actor Critic
CCI	Commodity Channel Index
CR	Cumulative Return
DDPG	Deep Deterministic Policy Gradient
DJI	Dow-Jones industrial
DRL	Deep Reinforcement Learning
DSR	Differential Sharpe Ratio
MACD	Moving Average Convergence Divergence
MDD	Maximum Drawdown
OES	Original Ensemble Strategy
PPO	Proximal Policy Optimization
QP	Quadratic Programming
RSI	Relative Index Strength
S&P	Standard & Poor's
SR	Sharpe Ratio