TÉCNICO LISBOA

Advantage Actor-Critic Algorithm Application to Pairs Trading Strategy

Diogo Emanuel Graça Rodrigues

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor(s): Prof. Nuno Cavaco Gomes Horta

Examination Committee

Chairperson: Prof. Teresa Maria Sá Ferreira Vazão Vasques
Supervisor: Prof. Nuno Cavaco Gomes Horta
Member of the Committee: Prof. Helena Isabel Aidos Lopes Tomás

January 2021

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

The first word of thanks must go to my thesis supervisor, professor Nuno Horta, for his patience, dedication and constant availability during this work. His feedback was fundamental for the best way to guide this thesis. All the best for the future!

A word of appreciation to my family who is my foundation in everything. A special thanks to my parents and brother. Finally, I would like to mention my friends, who no matter what happens or how much time I may not spend with them, they are always there to support me.

I can say that I am a lucky person!

Resumo

Pairs trading é uma estratégia bem conhecida de mercado neutro. Na compra de um título financeiro com um preço relativamente mais baixo em comparação com a sua contraparte no par, e, consequentemente, a venda a descoberto do outro constituinte do par, o investidor pode lucrar quando o preço do par converge. Nos primórdios, *pairs trading* era bastante popular devido às oportunidades de obtenção de lucro por arbitragem. No entanto, com tantos investidores, incluindo fundos de investimento, a procurar estas oportunidades de arbitragem, a rentabilidade desta estratégia começou a deteriorar-se. Neste estudo propomos um método alternativo de extração do sinal de negociação, TLS, superior ao método tradicional, OLS, em todas as condições. Além disso, demonstramos as implicações que a escolha do tamanho da janela de negociação pode ter na rentabilidade da estratégia. Com estas duas caraterísticas em consideração, investigamos uma nova abordagem à estratégia tradicional utilizando aprendizagem por reforço profundo - particularmente com o algoritmo *advantage actor-critic*. Desenvolvemos o sistema de negociação, dando recompensas positivas ao agente por decisões adequadas e recompensas negativas por decisões erradas. Dado o sinal de negociação, o agente é treinado para selecionar as melhores decisões que maximizam a soma dos lucros futuros esperados. Os pares são selecionados entre 208 ETFs relacionados com *commodities*. As simulações consideram custos de transação e são realizadas ao longo de vários períodos entre janeiro de 2011 e dezembro de 2019. Os nossos resultados demonstram sucesso na aprendizagem do modelo proposto e apresentam possibilidades de extensão a outras aplicações financeiras computacionais.

Palavras-chave: *Pairs Trading*, Mercado Neutro, *Total Least Squares*, Aprendizagem por Reforço, Aprendizagem Profunda, *Actor-Critic*

Abstract

Pairs trading is a well known market-neutral strategy. By buying a stock with a relatively low price compared to its counterpart in the pair, and accordingly short sells the other one, a profit can be expected when the pair's price converges. In the early days, pairs trading strategy was quite popular due to the opportunities to obtain arbitrage profit. However, with so many investors, including hedge funds, sought these arbitrage opportunities, its profitability began to deteriorate. In this study, we propose an alternative method of extracting the trading signal, TLS, superior to the traditional method, OLS, under all conditions. Also, we demonstrate the implications that the trading window size choice may have on the strategy's profitability. With these two features in consideration, we research a novel approach to traditional pairs trading strategy using deep reinforcement learning - particularly with the advantage actor-critic algorithm. We develop the trading system by giving positive rewards to the agent for appropriate decisions and negative rewards for wrong decisions. Given the trading signal, the agent is trained to select the optimal trading decisions to maximise the expected sum of discounted future profits. Pairs are selected from 208 commodity-linked ETFs. The simulations take into account transaction costs and are conducted over several periods between January 2011 and December 2019. Our results demonstrate success in learning the proposed model and present possibilities for an extension to other computational finance applications.

Keywords: Pairs Trading, Market Neutral, Total Least Squares, Reinforcement Learning, Deep Learning, Actor-Critic

Contents

List of Tables

List of Figures

List of Acronyms

A2C Advantage Actor-Critic

A3C Asynchronous Advantage Actor-Critic

ADF Augmented Dickey–Fuller

AI Artificial Intelligence

CAP Credit Assignment Problem

DP Dynamic Programming

DQN Deep Q-Networks

DRL Deep Reinforcement Learning

EIV Errors-in-Variables

ETF Exchange-Traded Fund

FWER Family-Wise Error Rate

GA Genetic Algorithms

GHE Generalised Hurst Exponent

GPI Generalised Policy Iteration

MC Monte-Carlo

MDD Maximum Drawdown

MDP Markov Decision Process

ML Machine Learning

OLS Ordinary Least Squares

ReLU Rectified Linear Unit

RL Reinforcement Learning

ROCC Return on Committed Capital

ROI Return on Investment

SR Sharpe Ratio

SSD Sum of Euclidean Squared Distance

TD Temporal-Difference

TLS Total Least Squares

Chapter 1

Introduction

1.1 Pairs Trading Overview

Practice often shows that profitable investment strategies do not have to be complicated. The perfect example of this is pairs trading, where the beauty is in its simplicity.

Pairs trading is an important research area of computational finance that typically relies on time series data of securities for investment [1], where securities are bought and sold in pairs for arbitrage opportunities. It has been employed as one important long/short[1] equity investment tool by hedge funds and institutional investors for decades [2]. Around 1950, Alfred Winslow Jones used the idea of trading pairs by buying in specific stocks and selling in others [3]. The concept as we know it today was developed in the 1980s. In the early days, pairs trading methods were quite popular due to the opportunities to obtain arbitrage profit [4–6]. However, with so many investors, including hedge funds, sought these arbitrage opportunities, its profitability began to deteriorate [7]. Recently, significant research has been conducted to overcome these shortcomings in the strategy [8–12].

The classic form of pairs trading is based on finding two securities with some relation[2] and trying to take advantage of their prices differences. The strategy buys the security with a relatively low price compared to its counterpart in the pair, and accordingly short sells the other one and expects the prices of the securities in the pair would converge within the intended time horizon. The price gap of the two securities, also known as the spread[3], acts as a signal to open and close positions in the pair's constituents. During the trading period, a position is opened when the spread widens by a certain threshold, and afterwards, the position is closed when the spread reverts. Statistically, spreads made by two securities have a mean-reversion in the long run [4]. Based on this criterion, it is possible to profit from this long/short strategy.

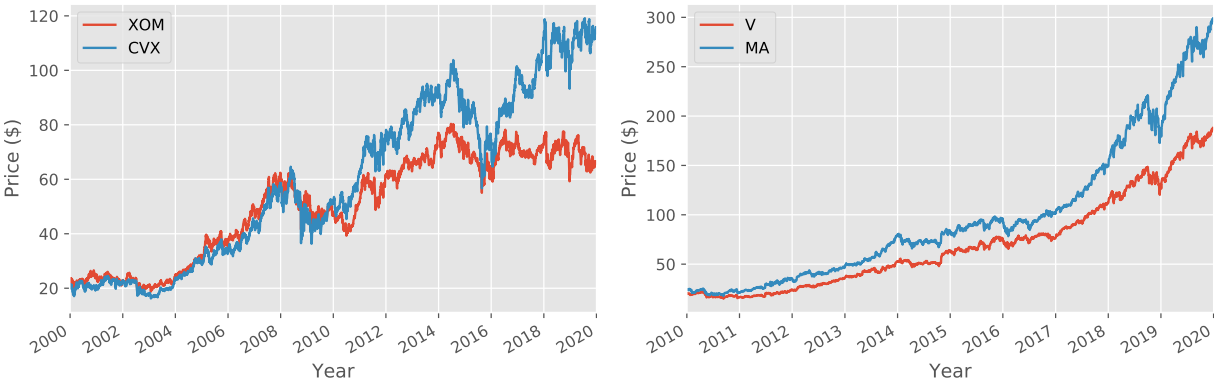
From the description presented above, we can divide the mechanism of pairs trading into two main

[1] In the financial markets nomenclature, a long position entails buying a security whereas a short position entails selling a security. More specifically, the latter means that an investor borrows a stock, sells the stock, and then repurchases the stock to return it to the lender.

[2] In the next chapter, we will analyse with detail the various possibilities for finding those securities.

[3] For now, the spread is defined as the difference between the two securities' price.

steps: identifying two securities[4] (stocks are probably the most straightforward example for the reader to think about) and trading itself. Regarding the first step, we are looking for two price series displaying similar behaviour or being linked to each other. Eventually, this means that the identified securities are exposed to related risk factors and tend to react identically. Figure 1.1 illustrates this behaviour in some top-rated stocks. Figure 1.1(a) shows how this can be seen in two of the most recognisable oil and gas supermajors[5]. Figure 1.1(b) presents an example with two of the world's largest payment processing networks. Two securities that verify an equilibrium relation between their prices series can form a pair.



(a) Price series from two oil and gas companies, Exxon Mobil Corporation (XOM) and Chevron Corporation (CVX).

(b) Price series from two of the most giant American multinational financial services corporations, Visa Inc. (V) and Mastercard Inc. (MA).

Figure 1.1: Price series from stocks which could potentially form profitable pairs.

Once a pair is identified, we move on to the second step. Here, the underlying premise is that if two securities that have moved together in the past, the same should persist in the near future. We are talking about a mean-reverting process, and the rationale behind it is that there is a long-term equilibrium (mean) for the spread. Thus, when irregularities occur, attractive trading opportunities can arise and, consequently, profit from its correction can be made. The spread between the pair's constituents must be continuously monitored to find such opportunities. When a statistical anomaly is detected, we enter a market position. This position is closed after an eventual spread correction. It is interesting to note that this strategy relies on the relative value of two securities, regardless of its absolute value.

To ensure a full understanding of the strategy, we will introduce it using an example of this work. A more formal description of the trading setup is available in section 2.2.1. For now, let us assume that two different securities were previously identified as forming a potential pair, in this case, AGQ and SLVO. Both are indexes that follow the behaviour of precious metals, namely silver. The connection between them is quite evident as they are exposed to the same sector. Figure 1.2 illustrates how its price series behave in a very similar way. Note that, price series have been normalised[6], a widespread technique when we inspect more than one security at the same time.

[4]Meanwhile, we will assume that each pair is only made up of two elements (univariate). In the literature, we can find examples where each component has more than one element (multivariate) [13].

[5]Supermajor is a name used to describe the world's largest few privately-owned oil and gas companies.

[6]In this context, the concept of normalising means just dividing the entire price series by the first value, ensuring that both securities' price series starts in one unit.

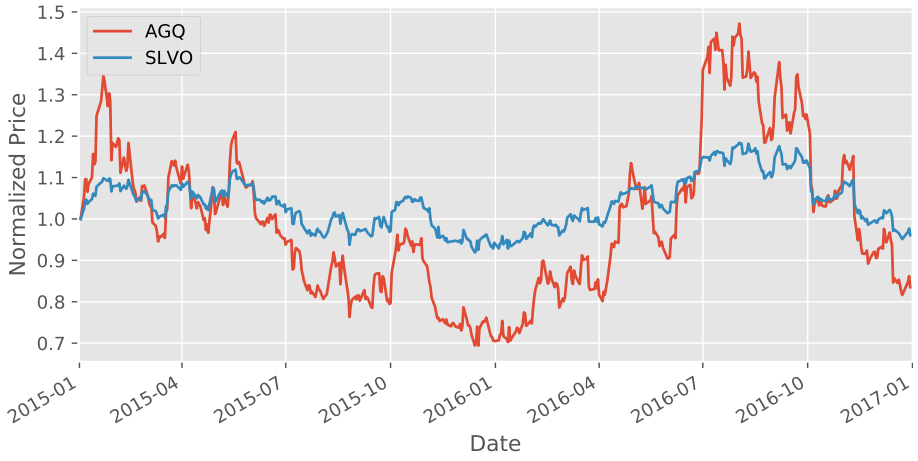


Figure 1.2: Formation period from 2015 to 2017.

The period shown in figure 1.2 is called the formation period, and it is used to select potential pairs. In this period, the investor calculates the spread's mean and standard deviation between the pair's two constituents. These values describe the statistical behaviour known to the pair and in which the investor expects to remain approximately constant in the near future (at least within the defined time horizon for trading).

In the following period (called the trading period), the spread, $S_t = SLVO_t - AGQ_t$, is normalised[7] and carefully monitored, as shown in figure 1.3. Although the signal always tends to return to the mean, there are some deviations. Depending on the magnitude of these deviations, a trade can be initiated. For this purpose, long and short thresholds are defined, which will define the minimum deviation required to open a long and short position, respectively. We are considering that the spread will rise in a long position, as we believe that its value is below expectations. So, it means buying SLVO and selling AGQ. In contrast, we are assuming that the spread will decrease in a short position, as its value is above our expectation and opposite transactions will occur. The positions are liquidated, that is, closed when the spread reverts to its expected value (mean).

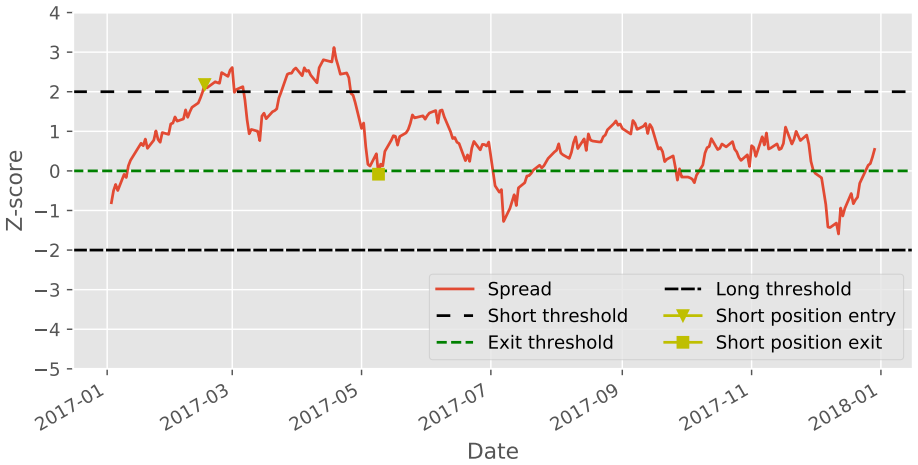


Figure 1.3: Example of the strategy in the trading period in 2017.

[7]Normalising the spread is the same as saying computing the Z-score values. In statistics, a Z-score (also called a standard score) is the number of standard deviations by which the value of a raw score is above or below the mean value. Formally, $Z = \frac{x-\mu}{\sigma}$, where x is the observed value, μ is the mean of the sample, and σ is the standard deviation of the sample.

Analysing figure 1.3, we can see a lucrative trading opportunity. When the spread signal exceeded the corresponding threshold, a short position was opened (marked by the triangle pointing down in yellow); and when the spread successfully returned to zero, the position was closed (marked by the square also in yellow).

The described strategy has several advantages. It goes beyond the arduous and very subjective process of evaluating a security, which is a fundamental step in deciding whether to buy or sell. By focusing only on the price difference between the securities - relative pricing - this problem is mitigated. Besides, pairs trading is a well-known market neutral trading strategy, enabling traders to virtually profit from any market conditions: uptrend, downtrend, or sideways movement.

1.2 Deep Reinforcement Learning Motivation

Deep reinforcement learning (DRL) is a combination of reinforcement learning (RL) and deep learning. It is one of the topics that has dominated the artificial intelligence (AI) world in recent years with its psychology-based model, mimicking how humans learn and improve on a task.

The intersection of various fields of science is a unique RL characteristic. There is a branch of many different fields of endeavour trying to study the same problem as RL. So what is that problem? It is the science of decision-making. This comprehensiveness makes RL so general and interesting across many different fields. The Venn diagram in figure 1.4 represents the major science fields trying to understand the optimal way to make decisions.

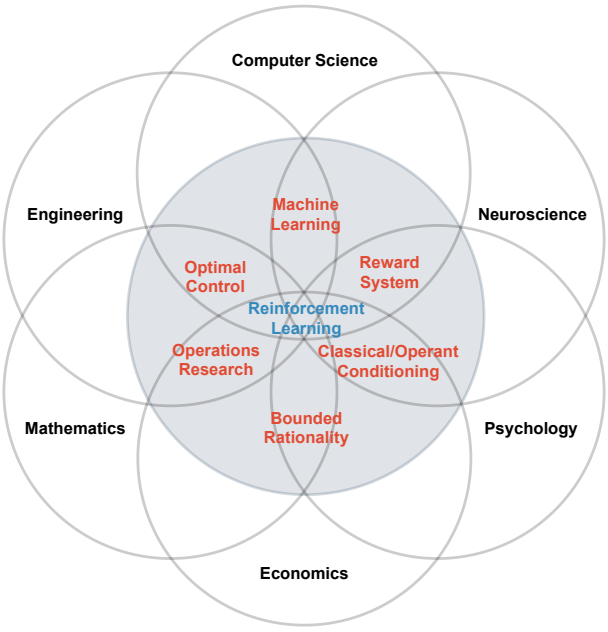


Figure 1.4: Many faces of reinforcement learning. Adapted from: [14].

Focusing on computer science, specifically machine learning (ML), it can be roughly divided into three branches: supervised, unsupervised and reinforcement learning. The main goal of supervised learning is to predict values or classes based on labelled data with the desired output and deducing the

relationship between them to generate accurate outputs for further examples. Unsupervised learning deals with clustering and finding relations in unlabeled data. Reinforcement learning is somewhat between supervised and unsupervised learning. RL deals with how some arbitrary being (formally referred to as an "agent") should act and behave in a given environment. The agent learns from experience and exploration, being goal-oriented. The desired behaviour is learned by trying to maximise the long-term reward it receives. Some significant differences that make reinforcement learning distinct from other machine learning paradigms are:

- There is no supervisor, only a reward signal.
- Feedback is delayed, not instantaneous.
- Time really matters (sequential, non-IID data[8]).
- Agent's actions affect the subsequent data it receives.

RL motivated many of the earliest computational studies of learning, but most researchers had gone to other things, such as pattern classification, supervised learning, and adaptive control [15]. However, lately, this field has evolved and matured in several directions and has gradually become one of the most active research areas in recent times that has made it very much cutting-edge.

In 2013, a company called DeepMind uploaded their pioneering paper [16]. This paper demonstrated how a computer learned to play Atari 2600 video games from the Arcade Learning Environment just by looking at the screen pixels and receiving rewards when the game score increased. It was a remarkable result because each game and goals were very different and designed to be challenging for humans. The developed model was applied in seven different games, with no adjustment of the architecture or learning algorithm, and surpassed a human expert on three of them. In February 2015, the paper [17] from the same company was featured on the cover of Nature, one of the most prestigious journals in science. This paper developed a model applied to 49 different games (again using the same algorithm, network architecture and hyperparameters) and achieved superhuman performance in half of them.

1.3 Objectives

Research on the field focuses mainly on traditional methods and statistical tools to improve the pairs trading strategy's critical aspects. Pairs selection has been the stage where the most effort has been devoted, with several studies focusing on the various existing statistical methods and their variations. Our motivation is then to improve the trading strategy.

In recent years, reinforcement learning is gaining momentum, namely when applied together with deep learning (deep reinforcement learning). Its disruptive capacity and the little application known to date in pairs trading specifically led us to study its application possibilities.

This work is the follow-up of Sarmento and Horta [12]. The standard bases were the dataset used (in our case, using the adjusted daily closing price), the criteria for selecting the pairs adopted, and the

[8]Not independent and identically distributed data.

trading simulation environment's construction. Nevertheless, it is necessary to emphasise that although it was the foundations, everything was built by us from scratch. The contributions of our work can be defined based on the two proposed study phases, described below:

1. A pairs trading strategy's success depends on finding the right pairs and extracting the key features from them. Here, we want to highlight two main features that we will conduct experiments and scrutinise their impact on the strategy. First, we used different spreads calculated using ordinary least squares (OLS) and total least squares (TLS) to see how the results differ depending on the spread used for input. Second, depending on the formation window and trading window, the spread and hedge ratio will be varied. Therefore, we propose to study a set of six window sizes for selecting the optimal window size, which had the best performance.

2. We investigate whether it is possible to train a deep reinforcement learning model that optimises pairs trading strategies beyond traditional methods. To this end, we propose a new method to optimise the pairs trading strategy using deep reinforcement learning, specifically advantage actor-critic, since pairs trading strategy can be thought of as a game. After opening a portfolio position, the profit can be set if the portfolio is closed in the correct timing. Hence, if we set this strategy as a game by setting the right positions optimised in spreads in trading windows, we can profit.

Together with these two main study phases, this work proposes to use and analyse the suitability of exchange-traded funds (ETFs) in a pairs trading setting. This evaluation can be seen as particularly relevant as most studies are done on a regular stock basis. Simulations are run over several periods from January 2011 to December 2019.

1.4 Thesis Outline

The rest of this thesis is organised as follows. Chapter 2 introduces the background and related work concerning pairs trading alongside the fundamental and mathematical concepts on the foundation of reinforcement learning. Chapter 3 describes the proposed model in this work. First, highlighting the two proposed methods for extracting the trading signal. Second, describing the framework for selecting pairs. Lastly, detailing our DRL model. Chapter 4 includes some practical information on how the investigation was conducted. Chapter 5 shows the results and provides a discussion of the experiments. Finally, chapter 6 provides some concluding remarks and a brief discussion of possible future work for developing the proposed system.

Chapter 2

Background and Related Work

This chapter introduces some fundamental concepts for the work, and a revision of the existing literature focused on the various fields. There is a clear division in this chapter, the section referring to the pairs trading strategy and the section referring to reinforcement learning. Regarding pairs trading and as already described in section 1.1, there are two major stages in the strategy: selecting the pairs and the trading itself. Following this line of reasoning, we will focus on the two stages separately. Finally, a more in-depth look into the world of reinforcement learning is presented.

2.1 Pairs Selection

Krauss [18] provides a comprehensive survey of pairs trading literature which clusters the techniques in the following categories: distance, cointegration, time-series, stochastic control, and other approaches. The distance and cointegration categories tend towards the pairs selection problem while time-series and stochastic control concern the trading strategy. The latter two usually assume that the pairs are chosen a priori and instead focus on defining/optimising the strategy execution parameters, such as trading boundaries, detection regime, and optimal trade allocations. In this section, we will focus more on the first two categories and selecting pairs.

2.1.1 Distance Approach

The most cited paper in pairs trading and the most prominent study for distance-based selection criteria is the work from Gatev et al. [4], hereafter GGR. They used normalised U.S. stock price data from 1962 to 2002 to test the profitability of pairs trading. The authors constructed a cumulative total return index for each stock and normalised to the first day of a 12-month formation period. Subsequently, they applied the sum of Euclidean squared distance (SSD) between price time series of all possible pair combinations ($\frac{n(n-1)}{2}$, with n being the number of stocks). Only the pairs yielding the minimum historic SSD are passed on to the trading period.

Do and Faff [7] reviewed and analysed the work of GGR to find diminishing profitability of distance approaches in recent years. A detail pointed out in this work is that pairs trading's profitability is much

more about identifying and excluding divergent pairs as it is about identifying and including convergent pairs.

Krauss [18] highlighted that the choice of SSD as a selection metric is analytically suboptimal. To demonstrate this statement, let $p_{i,t}$ and $p_{j,t}$ denote realizations of the normalised price processes $P_i = (P_{i,t})_{t\in T}$ and $P_j = (P_{j,t})_{t\in T}$ of securities i and j composing a pair. Empirical spread variance $s^2_{P_i-P_j}$ can be expressed as

$$s^2_{P_i-P_j} = \frac{1}{T}\sum_{t=1}^{T}(p_{i,t}-p_{j,t})^2 - \left(\frac{1}{T}\sum_{t=1}^{T}(p_{i,t}-p_{j,t})\right)^2. \tag{2.1}$$

We can solve for the average sum of squared distances $\overline{ssd}_{P_i,P_j}$ in the formation period,

$$\overline{ssd}_{P_i,P_j} = \frac{1}{T}\sum_{t=1}^{T}(p_{i,t}-p_{j,t})^2 = s^2_{P_i-P_j} + \left(\frac{1}{T}\sum_{t=1}^{T}(p_{i,t}-p_{j,t})\right)^2. \tag{2.2}$$

The "ideal pair", according to the SSD criterion, is the one that minimises (2.2). It is trivial to see that this would mean a spread of zero across the formation period. Logically, this is not consistent with the idea of potentially profitable pairs, because if there are no deviations of any kind (or of a shallow magnitude), there will be no trade opportunities. Thus, GGR's selection metric is likely to form pairs with low spread variance and limited profit potential.

We will include the work of Chen and Li [19] in this section because despite using Pearson correlation on return level for identifying pairs, the authors used the same dataset and time frame as GGR. The authors intend to build a more robust selection metric that can find promising pairs without penalising individual price differences, which was the case in GGR. First, let us consider sample variance of spreads returns, defined as return on buy minus return on sell,

$$s^2_{R_i-R_j} = s^2_{R_i} + s^2_{R_j} - 2\hat{\rho}_{R_i,R_j}\sqrt{s^2_{R_i}}\sqrt{s^2_{R_j}}. \tag{2.3}$$

We can immediately infer from (2.3) that imposing high return correlation $\hat{\rho}_{R_i,R_j}$ between assets i and j leads to lower variance of spread returns. However, the return time series of the individual securities may still exhibit vastly different variances. That said, we can already say that this selection metric is more flexible than minimising SSD.

Results using Pearson correlation reported average monthly raw returns of 1.70%, almost twice as high as those of GGR. Although the results presented by the return correlation are more favourable than with SSD, they continue to be far from optimal in this empirical point of view. Two securities correlated on return level, it does not necessarily mean that they share an equilibrium relationship, and there is no theoretical foundation that divergences are reversible. With this in mind, many correlations in this study may well be spurious, so a new approach is needed. In the next section, we will introduce cointegration.

2.1.2 Cointegration Approach

Before reviewing the existing literature on this approach, let us define cointegration, a fundamental concept from now on. The cointegration concept was an innovative mathematical model in economics developed by Nobel laureates Engle and Granger [20]. A set of variables is said to be cointegrated if a linear combination of those variables has a lower order of integration. In other words, cointegration exists if a set of $I(1)$ variables (non-stationary) can be used to model an $I(0)$ variable (stationary). Formally, assume that x_t and y_t are two non-stationary time-series, so both $I(1)$, cointegration implies that there is a parameter β such that the following equation is a stationary[1] process,

$$y_t - \beta x_t = \mu + \varepsilon_t, \tag{2.4}$$

where μ is the mean of the cointegration model. ε_t is a stationary, mean-reverting process and is referred to as residual cointegration. The parameter β is known as the cointegration factor. Cointegration is particularly interesting because it provides a way to create a stationary time series artificially, that can be used for trading. Finding a raw financial time series with this property is an arduous task and is rarely found. Price series are often non-stationary, and thus, in theory, completely unpredictable.

Testing for cointegration identifies stable, long-run equilibrium between sets of variables. Engle-Granger two-step method and the Johansen's method are the most widely used to test for cointegration.[2] In the two-step Engle-Granger method, we first set up a cointegration regression between the series, as stated in (2.4), using linear regression, and then save the residuals, ε_t. Subsequently, we test the residual regression ε_t to determine whether it is a spurious regression or stationary.

The most popular stationary test in cointegration is the Augmented Dickey-Fuller (ADF) test. The ADF test is a hypothesis test designed to assess the null hypothesis that a unit root is presented in a time series, proposed by Dickey and Fuller [22]. The test assumes stationarity as the alternative hypothesis. A consecutive change in time series can be modelled as

$$\Delta y_t = \alpha + \beta t + \lambda y_{t-1} + \sum_{i=1}^{p-1} \delta_i \Delta y_{t-i} + \mu_t, \tag{2.5}$$

where $\Delta y_t \equiv y_t - y_{t-1}$, α is a constant, β is the coefficient on a time trend, p is the lag order of the autoregressive process, μ_t is an error term and serially uncorrelated. Imposing the constraints $\alpha = 0$ and $\beta = 0$ corresponds to modelling a random walk[3], and using the constraint $\beta = 0$ corresponds to modelling a random walk with a drift.

The unit root test is carried out under the null hypothesis $H_0 : \lambda = 0$ against the alternative hypothesis $H_1 : \lambda < 0$. If the hypothesis is rejected, the change of a time series at time t depends on the series's value at time $t-1$, meaning that the series cannot be a simple random walk. The test statistic associated is the regression coefficient $\hat{\lambda}$ (calculated with y_{t-1} as the independent variable and Δy_t as

[1] Stationarity is a property of a stochastic process. Intuitively, it means that the statistical properties of a process generating a time series do not change over time. It does not mean that the series does not change over time, just that the way it changes does not itself change over time. In other words, a stochastic process is stationary if its mean and variance are constant over time.

[2] We will focus on the two-step Engle-Granger method, so readers interested in Johansen's method may refer to [21].

[3] A random walk is a stochastic process that describes a path that consists of a succession of random steps on some mathematical space.

the dependent variable) divided by the standard error (SE) of the regression fit, $\frac{\hat{\lambda}}{\text{SE}(\hat{\lambda})}$. This value is then compared with the relevant critical values corresponding to the test statistic distribution, which can be found tabulated and used to decide whether the hypothesis can be accepted or rejected at a given probability level.

In short, the cointegration approach allows selecting pairs whose constituents are cointegrated. If two securities, X_t and Y_t are found to be cointegrated, then, by definition, the resulting series from the linear combination,

$$S_t = Y_t - \beta X_t, \tag{2.6}$$

where β is the cointegration factor, must be stationary. Defining the spread series in this way is very convenient, as, in these conditions, it is expected to be mean-reverting, and can be used as a trading signal.

In pairs trading, the cointegration factor is common called hedge ratio, and it describes the amount of Y to purchase or sell for every unit of X. The hedge ratio can refer to a dollar value of security Y, or the number of units of security Y, depending on the approach taken. Further analysis of this distinction is detailed in section 2.2.1.

Vidyamurthy [23] provides the most cited work for cointegration-based pairs trading. Pairs are pre-selected through a screening process based on statistical or fundamental similarity measures before testing for cointegration. The screening has a dual purpose of domain reduction for expensive cointegration tests and a latent selection bias towards inter-industry pairs. Despite lacking empirical application, his framework lays the foundation for all further cointegration-based pairs trading studies.

Rad et al. [24] provide the first large-scale implementation of the cointegration approach on normalised data backtested from 1962 until 2014, following GGR and Vidyamurthy [23]. First, the authors identify the stocks with minimum SSD in a 12-month formation period. Next, they apply the Engle-Granger cointegration test to these stocks and only retain the top 20 stocks of the SSD ranking that are also cointegrated. As noted by Krauss [18], the SSD sorting heuristic introduces selection bias, making only the smallest SSD pairs reach the cointegration test. This distance constraint reduces spread variance so the procedure could not be selecting the most profitable cointegrated pairs.

Mikkelsen [25] compared the distance and cointegration approaches for both high-frequency and daily data to investigate the profitability of pairs trading for Norwegian seafood companies. The performance of both approaches was similar. Conversely, Huck and Afawubo [26] showed that selected pairs based on cointegration more often exhibit mean-reverting behaviour than distance pairs. The authors also showed how pairs formed purely on cointegration yield higher spread volatility and are more profitable. Clegg and Krauss [27] relax the stationary condition for cointegration to develop the concept of partial cointegration. In their analysis, the developed model outperforms the benchmark model of GGR and two other cointegration variants.

Following Krauss [18], we can conclude that the cointegration approach is more rigorous than the distance approach. The main reason is that cointegration identifies econometrically more sound equilib-

rium relationships.

2.1.3 Other Approaches

Although these two approaches discussed in detail are the most commonly mentioned in the literature, there are, of course, other methodologies to address this problem. We want to highlight two of them. The first has to do with the use of evolutionary algorithms in finance. Aguilar-Rivera et al. [28] provide a recent review of the current literature space of applying evolutionary computation methods to solving financial problems. Goldkamp and Dehghanimohammadabadi [13] suggests the reformulation of the optimisation for the pairs trading by introducing the nondominated sorting genetic algorithm II (NSGA-II). Finally, it is relevant to highlight the work of Gupta and Chatterjee [29], who incorporated the lead-lag relationship between the chosen stocks to select the best pairs for trading. The authors suggest that when the proposed measure is clubbed with SSD measure, i.e., identifying the pairs through the optimisation of these two measures, this new selection metric consistently generated the pairs with the best profits compared to the other measures.

Table 2.1 summarises what we consider to be the most relevant work in pairs selection. It presents the sample considered by each work and the main contribution to the field.

Table 2.1: Literature exploring pairs selection techniques.

Article	Sample	Contribution
Gatev et al. [4]	U.S. CRSP[4] 1962-2002	Baseline approach in U.S. equity markets: pairs trading is profitable; returns are robust
Do and Faff [7]	U.S. CRSP 1962-2009	Expanding on GGR: profitability is declining and not robust to transaction costs; improved formation based on industry, number of zero crossings
Goldkamp et al. [13]	U.S. S&P 500 2011-2016	Improvements: multi-objective and multivariate pairs trading variants outperform single objective and univariate counterparts
Chen and Li [19]	U.S. CRSP 1962-2002	Improvements: correlation-based formation outperforms SSD rule
Vidyamurthy [23]	-	Most widely cited cointegration-based concept
Rad et al. [24]	U.S. CRSP 1962-2014	Expanding on GGR and Vidyamurthy [23]: first large-scale implementation of the cointegration approach
Mikkelsen [25]	Norwegian seafood co. 2005-2014	Comparison studies: comparing univariate pairs trading strategies - most notably distance versus different cointegration approaches
Huck and Afawubo [26]	U.S. S&P 500 2000-2011	*similar to the previous row. In addition, it compares the performance of high-frequency and daily data.
Clegg and Krauss [27]	U.S. S&P 500 1989-2015	Improvements: partial cointegration identifies promising pairs and generates good buy and sell signals
Gupta and Chatterjee [29]	(i) DJIA (ii) Sensex 30 (iii) Topix 30 2008-2016	Improvements: incorporation of lead-lag relationship between stocks

[4]The Center for Research in Security Prices (CRSP) is a provider of historical stock market data.

2.2 Trading Strategy

After exploring the most relevant techniques for selecting pairs, we will examine the existing strategies for trading execution. Unlike the selection stage, where we can find different configurations, the trading strategy is usually based on the threshold trading model described next.

2.2.1 Threshold-based Trading Model

The strategy proposed by GGR is the traditional framework for trading execution. After the spread series between the two constituents of the pair is defined, it is converted into a Z-score[5] used as a trading trigger. The criterion for opening a position is based on the trading signal divergence. When the trading signal diverges by more than two historical standard deviations, a position is opened. The position is closed when it converges again to the mean. Furthermore, if a position has been opened, but the trading signal has not returned to the mean during the trading window, the portfolio position is closed by "force" and losses may occur. This description should be familiar to the reader, as it was the implementation introduced in section 1.1.

Figure 2.1 exemplifies how this strategy proceeds and was taken from the work of GGR. The illustration uses two stocks, Kennecott and Uniroyal, in the trading period defined by the authors. The top two lines represent the normalised price paths with dividends reinvested, and the bottom line indicates whether the daily position is open or closed. We can see that figure 2.1 illustrates the description above very well since a position is opened when there is a more significant divergence in the price between the two stocks and closed when it converges.

Figure 2.1: Threshold-based trading model. Source: [4]

In case two securities X and Y have already been identified as forming a potential pair, this model can be described more formally as follows:

1. Calculate the spread's ($S_t = Y_t - X_t$) mean, μ, and standard deviation, σ, during the pair's formation period.

2. Define the trading boundaries/thresholds: the threshold that triggers a long position, the threshold that triggers a short position and the threshold that defines when the position should be closed.

[5] Recall that a Z-score is calculated by subtracting the mean of a group of values from an individual raw score and dividing the difference by the standard deviation of that same group.

3. Convert the spread into a Z-score and monitor its evolution to detect when a threshold is crossed.

4. If the long threshold is crossed, buy Y and sell X. If the short threshold is crossed, sell Y and buy X. In the case of an active position and the exit threshold is triggered, close the position.

It is important to note that the spread's definition is tied to the technique used to identify the pairs. If we follow the example above, using the distance approach, the spread is defined by $S_t = Y_t - X_t$. In the case of using the cointegration approach, the spread is defined by $S_t = Y_t - \beta X_t$. Subsequently, this will also affect the way positions are set. In the case of the spread defined as $S_t = Y_t - X_t$, open a long position entails buying Y and selling X in the same number of shares or investing the same capital on both if the investor follows a dollar-neutral position. In the case where the spread is defined as $S_t = Y_t - \beta X_t$, there seems to be no consensus in the literature in the most appropriate way to deal with the hedge ratio[6]. According to Chan [30], the investor should invest in one share of Y and β shares of X. Rad et al. [24] state that as by definition, long and short positions are not valued equally, so the authors suggest investing \$1 in the long leg[7] and determine the value in the short leg according to the hedge ratio. However, other authors decide to completely ignore the hedge ratio to enforce a money-neutral position and invest the same amount of money in each leg, as was the case with Rudy et al. [31]. It is also essential to leave a final note regarding these methodologies. In practice, affirming that positions are money neutral on both sides of the trade is not always possible, as an investor, typically, cannot buy share fractions. The same reasoning can be applied in the case where we take into account the hedge ratio. One way to mitigate this problem is to increase the amount of money invested so that it is always possible to find a common multiple.

2.2.2 Alternative Trading Models in the Literature

The formerly described strategy is far from perfect. It has no concern with optimising entry points, and there is no guarantee that the moment when the threshold is exceeded is the ideal time for entering a position. The spread may continue to diverge before converging, and we can see the value of our portfolio declining.

Some efforts to create more robust models have emerged, with emphasis on the fields of stochastic control theory and machine learning, in particular time series forecasting. Some examples in the literature include Elliott et al. [5], who describe the spread with a mean-reverting Gaussian Markov chain, observed in Gaussian noise. The trading positions are set according to the relative difference between the series prediction and the observation. Moreover, Mudchanatongsuk et al. [8] optimised a pairs trading system as a stochastic control problem. They modelled the log-relationship between a pair of stock prices as an Ornstein-Uhlenbeck process and tested their model with simulated data; the results showed that their strategy performs well. Also, Holý and Tomanová [32] suggested that high-frequency data are contaminated by the market microstructure noise, which causes significant bias in parameter estimation when not taken into account. Therefore, the authors proposed an Ornstein-Uhlenbeck process robust

[6] Recall that in pairs trading, the hedge ratio comes from the cointegration factor.
[7] Sometimes in the literature, each constituent of a pair is called a pair's leg.

to the noise and perform better than tradition estimators such as ARMA(1,1) and maximum likelihood. Huang et al. [33] use genetic algorithms (GA) to simultaneously optimise capital allocation between a small set of candidate pairs and trading signals in an overall mean-reverting trading system. According to the authors, the proposed method can generate robust models able to tackle the dynamics characteristics of the pairs trading application. Bayram et al. [34] proposed a different approach, where they integrated the expert opinions extracted from the Bloomberg database into a fuzzy decision-making process. According to the authors' analysis, the improvements developed through this fuzzy technique are even more remarkable when considering transaction costs.

As it was done in the previous stage, table 2.2 groups the trading models mentioned in this section and considered a more prominent contribution to pairs trading strategy.

Table 2.2: Literature focused on developing the trading strategy model.

Article	Sample	Contribution
Gatev et al. [4]	U.S. CRSP 1962-2002	Baseline approach in U.S. equity markets: standard threshold-based trading model
Elliott et al. [5]	Simulated data	Modelling the spread in state-space at the return level
Mudchanatongsuk et al. [8]	Simulated data	Derivation of the optimal strategy for a risky asset following an Ornstein-Uhlenbeck process under various utilities
Holý and Tomanová [32]	7 Big Oil stocks NYSE 2015-2018	Introduction of noise-robust estimators of the Ornstein–Uhlenbeck process parameters
Huang et al. [33]	10 semicondutor stocks TWSE 2003-2012	Presentation of a GA-based methodology for the optimisation of the trading model parameters
Bayram et al. [34]	NASDAQ energy sector 2013-2014	Introduction of expert opinions in a fuzzy logic system

2.3 Reinforcement Learning

After the two fundamental steps in the strategy of pairs trading have been detailed, it is time to investigate RL's field. First, let us overview the most relevant concepts and methods that can be found in the literature. Then, we will review the literature that applies RL to financial markets.

RL's essence is to learn by interacting and receiving feedback in the form of a reward signal. The agent interacts with the environment, and by observing the consequences of his actions, he can learn to change his behaviour taking into account the rewards he received. This paradigm of learning by trial and error, and interacting with the environment to achieve goals, makes RL of all ML forms the closest to how humans and animals learn. Initially, biological learning systems inspire many of the core RL algorithms [15]. Another key influence on RL is optimal control, which lent the mathematical foundations (mainly dynamic programming [35]) that underpin the field.

As we said before, RL's main idea is to learn through interaction to achieve a goal. The problem setup contains the learner and decision-maker, called the agent. It interacts with the environment, which includes everything outside the agent. This interaction is continuous, with the agent selecting actions and the environment responding to these actions presenting new situations to the agent. The environment also generates rewards, which are particular numerical values that the agent seeks to maximise over time through the choice of his actions. This idea is represented in figure 2.2.

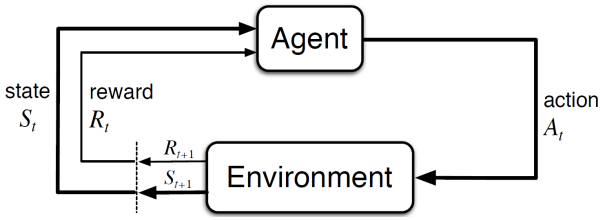


Figure 2.2: The RL framework. At time t, the agent observes a state s_t inherent to the environment. Given that observation takes action and transition to a new state, s_{t+1}, based on the current state and the action taken. Besides, a reward is also given to an agent, taking into account the decision he made. Source: [15].

Before going any deeper into what RL does, we need to introduce additional terminology.

2.3.1 States and Observations

A state s is a complete description of the world, i.e., no information about the world is hidden from the state. An observation o is a partial description of the state, which may omit information. A fully observed environment is when the agent can observe the complete state of the environment. A partially observed environment is when the agent can only see a partial observation.

2.3.2 Action Spaces

Depending on the environment, different kinds of actions are allowed. The set of all valid actions is usually called the action space. Some environments, such as the Atari and Go games, merely allow discrete actions, where only a finite number of actions is allowed. Other environments, such as those where the agent controls a robot in the physical world, have continuous action spaces. Here, actions are real-valued vectors. This distinction has significant consequences when choosing the DRL method. Some families of algorithms can only be applied directly in one case, and have to be substantially reworked for the other.

2.3.3 Policies

A policy fully defines an agent's behaviour; it is like a rule used by the agent to decide what actions to take, $a_t = \pi(s_t)$. As in real life, not everything is absolute, and our policy can be deterministic or stochastic. For a stochastic policy, the result is a distribution of actions given states,

$$
\begin{aligned}
a_t &\sim \pi\left(.|s_t\right) \\
\pi\left(a|s\right) &= \mathbb{P}\left[A_t = a|S_t = s\right].
\end{aligned}
\tag{2.7}
$$

In DRL, we deal with parameterised policies, whose outputs are computable functions that depend on a set of parameters (e.g. the weights and biases of a neural network) which we can adjust to change the behaviour via some optimisation algorithm. We often denote the parameters by θ or ϕ, so we usually write policies as $a_t = \pi_\theta\left(s_t\right)$ and $a_t \sim \pi_\theta\left(.|s_t\right)$.

2.3.4 Trajectories

A trajectory[8] τ is a sequence of states and actions with horizon H,

$$
\tau = \left(s_1, a_1, s_2, a_2, \ldots, s_H, a_H\right). \tag{2.8}
$$

The natural laws of the environment govern state transitions (what happens to the world between the state at time t, s_t, and the state at $t+1$, s_{t+1}), and depend on only the most recent action, a_t. It is a function that represents the system's dynamics and maps a state-action pair at time t onto a distribution of states at time $t+1$,

$$
\mathcal{P}^a_{ss'} = \mathbb{P}\left[S_{t+1} = s'|S_t = s, A_t = a\right]. \tag{2.9}
$$

2.3.5 Reward and Return

The reward function is critically important in RL because it is the feedback by which we measure the success or failure of an agent's actions in a given state. It depends on the current state of the world and the action just taken, and the next state of the world,

$$
R_{t+1} = R\left(s_t, a_t, s_{t+1}\right). \tag{2.10}
$$

The goal of the agent is to maximise some notion of cumulative reward. The most common kind of return is the infinite-horizon discounted return which is the sum of all rewards obtained from time-step t but discounted by how far off in the future they are obtained. Its formulation is given by

$$
G_t = R_{t+1} + \gamma R_{t+2} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}. \tag{2.11}
$$

Sometimes, it is possible to use a finite-horizon undiscounted return (which is just a regular sum of rewards obtained) - an example is if all sequences terminate. However, this is not usually the rule, and there are several reasons to use the discount factor $\gamma \in [0, 1]$:

- In RL, the reward is delayed, so the agent needs to determine what actions led to a specific outcome. This problem is known as the (temporal) credit assignment problem (CAP) (discussed in

[8] A trajectory is also frequently called episode or rollout.

[36] by Marvin Minsky in 1961) and tries to be solved using the discount factor.

- It is both intuitively appealing and mathematically convenient. On an intuitive level, cash now is better than cash later. Mathematically, prevents an infinite-horizon sum of rewards from going to infinity.[9]
- Animal/human behaviour shows a preference for immediate reward[10] and, as already mentioned in section 1.2, the core idea of many RL algorithms is based on biological learning systems.

2.3.6 Reinforcement Learning Problem

Whatever the choice of the return measure (whether finite-horizon undiscounted or infinite-horizon discounted), and whatever the choice of policy, RL's goal is to select a policy that maximises expected return when the agent acts according to it. The central optimisation problem in RL could then be expressed by

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi} \left[G_t \,|\, S_t = s \right], \tag{2.12}$$

with π^* being the optimal policy.

2.4 Markov Decision Processes

So far, we have informally discussed the agent's environment, so it is time to go formal. In most AI topics, we create mathematical frameworks to tackle problems. For RL, the answer is Markov decision processes (MDPs). The name of MDPs comes from the Russian mathematician Andrey Markov as they are an extension of the Markov chains. MDP provides an easy framework for modelling a complex problem.

An MDP is defined by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ where:

- $\mathcal{S}$ is a finite set of states;
- $\mathcal{A}$ is a finite set of actions;
- $\mathcal{P}$ is a state transition probability, $\mathcal{P}^a_{ss'} = \mathbb{P}\left[S_{t+1} = s' | S_t = s, A_t = a\right]$;
- $\mathcal{R}$ is a reward function, $\mathcal{R}^a_s = \mathbb{E}\left[R_{t+1} | S_t = s, A_t = a\right]$;
- γ is a discount factor where $\gamma \in [0, 1]$.

The name Markov decision process refers to the fact that the system obeys the Markov property, which refers to the fact that the future only depends on the current state, not history. The formal definition says that a state s_t is Markov if and only if

[9] A simple way to understand this is that the sum of the rewards is bounded to be smaller than a geometric series and the sum of a geometric series is finite as long as the absolute value of the ratio is less than 1, which is our case. In short, we have $\sum_{t=0}^{\infty} \gamma^t R_t \leq \sum_{t=0}^{\infty} \gamma^t R_{max} = \frac{R_{max}}{1-\gamma}$.

[10] We are fighting against delayed gratification here. A viral study became known as the "marshmallow experiment" [37], which describes this problem entirely.

$$\mathbb{P}\left[s_{t+1}|s_t\right] = \mathbb{P}\left[s_{t+1}|s_1, \ldots, s_t\right]. \tag{2.13}$$

In other words, the future and the past are conditionally independent given the present, as the current state encapsulates all the statistics we need to decide the future.

2.5 Value Learning

In the optimal policy of (2.12), there are three main domains to solve this problem. For example, we can:

- Use the model to find the sequence of actions that has the maximum reward. The model is derived from the transition function that simulates the environment without interacting directly with the environment, that is, to predict the next state given agent's action. The transition model has a fundamental role in what we call model-based learning. However, this falls outside the scope of this work, and from now on, we will focus on model-free methods.
- Analyse how good it is to reach a particular state or take a specific action, i.e. value learning.
- Derive the policy directly in order to maximise the rewards, called the policy gradient.

We will now focus on value learning methods. They try to find a way to answer how good it was to reach a particular state or how good it was to take a specific action. Therefore, value function methods are based on estimating the value (expected return) of a given state or state-action pair and then act according to a particular policy π forever after. There are four main functions of note here.

1. The on-policy state-value function $V^{\pi}(s)$, which gives the expected return starting from state s, and always act according to policy π,

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left[G_t \,|\, S_t = s\right]. \tag{2.14}$$

2. The on-policy action-value function or quality function $Q^{\pi}(s, a)$, which gives the expected return starting from state s, taking an arbitrary action a (which may not have come from the policy π), and then forever after act according to policy π,

$$Q^{\pi}(s) = \mathbb{E}_{\pi}\left[G_t \,|\, S_t = s, A_t = a\right]. \tag{2.15}$$

3. The optimal state-value function $V^{*}(s)$, which gives the expected return starting from state s and always act according to the optimal policy in the environment,

$$V^{*}(s) = \max_{\pi} \mathbb{E}_{\pi}\left[G_t \,|\, S_t = s\right] = \max_{\pi} V^{\pi}(s). \tag{2.16}$$

4. The optimal action-value function $Q^*(s,a)$, which gives the expected return starting from state s, taking an arbitrary action a, and then forever after act according to the optimal policy in the environment,

$$Q^*(s,a) = \max_\pi \mathbb{E}_\pi \left[G_t \,|\, S_t = s, A_t = a \right] = \max_\pi Q^\pi(s,a). \tag{2.17}$$

2.5.1 The Optimal Q-Function and the Optimal Action

There is an essential connection between the optimal action-value function $Q^*(s,a)$ and the action selected by the optimal policy. The optimal policy in s will select whichever action maximises the expected return from starting in s. As a result, if we have $Q^*(s,a)$, we can directly obtain the optimal action a, via

$$a^*(s) = \arg\max_a Q^*(s,a). \tag{2.18}$$

2.5.2 Bellman Equations

All four of the value functions obey a particular self-consistency equation called Bellman equations [35].[11] The Bellman equations' basic idea is that the value function can be decomposed into two parts: the immediate reward expects to get, plus the successor's discounted value. Formalising, the Bellman equations for the on-policy state-value functions is

$$\begin{aligned} V^\pi(s) &= \mathbb{E}_\pi \left[G_t \,|\, S_t = s \right] \\ &= \mathbb{E}_\pi \left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots \,|\, S_t = s \right] \\ &= \mathbb{E}_\pi \left[R_{t+1} + \gamma \left(R_{t+2} + \gamma R_{t+3} + \ldots \right) \,|\, S_t = s \right] \\ &= \mathbb{E}_\pi \left[R_{t+1} + \gamma G_{t+1} \,|\, S_t = s \right] \\ &= \mathbb{E}_\pi \left[R_{t+1} + \gamma V^\pi(s') \,|\, S_t = s \right]. \end{aligned} \tag{2.19}$$

The on-policy action-value function can similarly be decomposed,

$$Q^\pi(s,a) = \mathbb{E}_\pi \left[R_{t+1} + \gamma Q^\pi(s',a') \,|\, S_t = s, A_t = a \right]. \tag{2.20}$$

The Bellman equations for the optimal value functions are

$$V^*(s) = \max_a \mathbb{E}_\pi \left[R_{t+1} + \gamma V^*(s') \,|\, S_t = s \right], \tag{2.21a}$$

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

[11] For a further intuition about Bellman equations, the reader should inspect appendix A.

The inclusion of the operator $\max$ in (2.21) reflects the fact that whenever the agents get to choose its action, to act optimally, it has to pick whichever action leads to the highest value. However, they are non-linear equations (due to the operator $\max$), so there is no closed-form solution in general. Nevertheless, we can solve these equations using iterative methods. In the case we have complete information about the environment, it becomes a planning problem that can be solved by dynamic programming [35]. However, most of the time, this is not the scenario, so we cannot directly solve the Bellman equations, even though they are the theoretical foundations of many RL algorithms.

2.6 Common Approaches

Like we said before, when the model is fully known, following Bellman equations, we can use dynamic programming (DP) to evaluate value functions and improve policy iteratively. This section will introduce the most common model-free approaches to solve the RL problem (finding the optimal policy).

2.6.1 Generalised Policy Iteration

To find the optimal policy π^*, we can use generalised policy iteration (GPI), where policy iteration consists of policy evaluation and policy improvement. Policy evaluation makes the value function consistent with the current policy, and policy improvement makes the policy greedy with respect to the current value function. We can prove that this process of policy iteration works and always converges to the optimality. Considering a deterministic policy, $a = \pi(s)$, we can improve the policy by acting greedily $\pi'(s) = \arg\max_{a \in \mathcal{A}} Q^{\pi}(s, a)$. The value of this improved π' is guaranteed to be better from any state s over one step,

$$\begin{aligned} Q^{\pi}(s, \pi'(s)) &= Q^{\pi}\left(s, \arg\max_{a \in \mathcal{A}} Q^{\pi}(s, a)\right) \\ &= \max_{a \in \mathcal{A}} Q^{\pi}(s, a) \geq Q^{\pi}(s, \pi(s)) = V^{\pi}(s). \end{aligned} \tag{2.22}$$

Instead of performing these steps separately to converge (as in policy iteration), generalised policy iteration allows you to merge the steps, favouring progress to be made quicker.

2.6.2 Monte-Carlo Methods

Monte-Carlo (MC) methods use the most straightforward idea: it learns directly from episodes of raw experience without modelling the environment dynamics. It estimates the expected return from a state by averaging the return from multiple rollouts of a policy. MC is a model-free method since there is no knowledge of MDP transitions or rewards. Due to these facts, pure MC methods can also be applied in non-Markovian environments. On the other hand, they can only be used in episodic MDPs, as a rollout has to terminate for the return to be calculated. To learn the optimal policy by MC, we iterate it by following a similar idea to GPI.

2.6.3 Temporal-Difference Learning

Similar to MC methods, Temporal-Difference (TD) Learning is model-free and learns from episodes of experience. However, TD learning can learn from incomplete episodes by bootstrapping, i.e., we can use our estimate's current values to improve our estimate.

The simplest TD learning algorithm, TD(0), follows the main idea of TD learning, that is to update the value function $V(s_t)$ towards an estimated return $R_{t+1} + \gamma V(s_{t+1})$ (known as "TD target"). To what extent we want to update the value function is controlled by the learning rate hyperparameter α,

$$V(s_t) \leftarrow V(s_t) + \alpha \left(R_{t+1} + \gamma V(s_{t+1}) - V(s_t) \right), \tag{2.23}$$

where $\delta_t = R_{t+1} + \gamma V(s_{t+1}) - V(s_t)$ is called the "TD error".

2.6.4 Monte-Carlo vs Temporal-Difference

The MC method is accurate. Nevertheless, for a stochastic policy or a stochastic model, every run may have different results. So the variance is high. TD considers far fewer actions to update its value since it can learn online after every step. So the variance is low. The main problem with TD learning is that their step updates, on the initial conditions of the learning parameters, are biased. If learning works as intended, then the bias will reduce asymptotically over multiple iterations. However, the bias can cause significant problems, especially for off-policy methods (e.g. Q-Learning) and using function approximators. That combination is so likely to fail to converge that it is called "the deadly triad" in [15]. In short, high bias gives wrong results, but high variance makes the model very hard to converge. In practice, we can get the best of both methods by combining TD learning and MC evaluation, as is done in the TD(λ) algorithm [15]. Similarly to the discount factor, the λ in TD(λ) is used to interpolate between MC evaluation and bootstrapping. This way, we can balance the bias and the variance which can stabilise the training.

2.6.5 Classic Algorithms

Similarly to (2.23), TD learning can be applied to action-value estimation, which is often the most frequent case, being the basis for algorithms such as Q-learning and the state-action-reward-state-action (SARSA).

Before highlighting the main differences between the two, it is necessary to introduce two concepts first: update policy and behaviour policy. Update policy is how the agent learns the optimal policy, and behaviour policy is how it behaves.

On the one hand, Q-Learning is an off-policy algorithm where the agent learns the optimal policy using absolute greedy policy and behaves using other policies such as ε-greedy policy[12]. On the other hand, SARSA is on-policy, as the agent learns optimal policy and behaves using the same policy.

[12]With probability $1 - \varepsilon$ choose the greedy action, and with probability ε choose an action at random.

2.7 Policy Gradient Methods

Policy gradients is a subclass of policy search methods. In policy search methods, we do not need to maintain a value function model but directly search for an optimal policy π^*; hence they are also called policy-based methods. Typically, a parameterised policy π_θ are updated to maximise the expected return using either gradient-based or gradient-free optimisation [38]. Neural networks that encode policies have been successfully trained using both methods. Gradient-free optimisation can effectively cover low-dimensional parameter spaces. However, despite some successes in applying them to large networks [39], gradient-based training remains the preferred method for most DRL algorithms, being more sample efficient when policies possess a large number of parameters. With that said, from now on, we are going to focus on gradient-based optimisation methods.

When constructing the policy directly, it is common to output parameters for a probability distribution. For continuous actions, this could be the mean and standard deviations of Gaussian distributions, while for discrete actions this could be the individual probabilities of a multinomial distribution. The result is a stochastic policy from which we can directly sample actions.

We wish to consider methods for learning the policy weights θ based on the gradient of some utility/performance measure, $U(\theta)$. These methods seek to maximise performance, so their updates approximate gradient ascent/descent in U. As usual for gradient descent/ascent, we have

$$\theta_{t+1} := \theta_t + \alpha \widehat{\nabla U(\theta_t)}, \tag{2.24}$$

where $\widehat{\nabla U(\theta_t)}$ is a stochastic estimate whose expectation approximates the gradient of the performance measure U regarding its argument θ_t.

Policy Optimisation

In policy optimisation, we consider control policy parameterised by parameter vector θ, and a utility/performance function $U(\theta)$, defined as follows:

$$U(\theta_t) = \max_\theta \mathbb{E}\left[\sum_{t=1}^{H} R(s_t) \mid \pi_\theta\right], \tag{2.25}$$

where $\pi_\theta(a \mid s)$ is a stochastic policy, that is, the probability of action a in state s.

Likelihood Ratio Policy Gradients

We want to compute the gradient $\nabla_\theta U(\theta)$ so that by applying gradient ascent/descent, we can improve the probabilities of better trajectories τ - (2.24). The likelihood ratio policy gradient can be used for this purpose. Before doing so, one must emphasise that likelihood ratio methods only change the probabilities of experienced paths, and further, these methods do not attempt to change the actions taken in a given path.

Recalling the definition of a trajectory in (2.8) and let the total return from trajectory τ be

$$R(\tau) = \sum_{t=1}^{H} R(s_t, a_t), \tag{2.26}$$

where $R(s_t, a_t)$ is a single rollout estimate of the action-value function $Q(s_t, a_t)$. Rewriting $U(\theta)$ we have that the expected reward is equal to the sum of the probability of the trajectory times the corresponding rewards,[13]

$$U(\theta) = \mathop{\mathbb{E}}_{\tau \sim P(\tau;\theta)} \left[\sum_{t=1}^{H} R(s_t, a_t); \pi_\theta \right] = \sum_{\tau} P(\tau;\theta) R(\tau). \tag{2.27}$$

Given this new notation, our goal is to find θ such that

$$\max_{\theta} U(\theta) = \max_{\theta} \sum_{\tau} P(\tau;\theta) R(\tau). \tag{2.28}$$

Since our main goal is to solve the optimisation problem of (2.28) using stochastic gradient ascent - (2.24) - we need to find the gradient of our utility/performance function $U(\theta)$ with regards to θ:

$$\begin{aligned}
\nabla_\theta U(\theta) &= \nabla_\theta \sum_{\tau} P(\tau;\theta) R(\tau) && \text{definition of } U(\theta) \text{ (2.27)} && (2.29)\\
&= \sum_{\tau} \nabla_\theta P(\tau;\theta) R(\tau) && \text{Leibniz integral rule: swap } \nabla \text{ and } \textstyle\sum && (2.30)\\
&= \sum_{\tau} \frac{P(\tau;\theta)}{P(\tau;\theta)} \nabla_\theta P(\tau;\theta) R(\tau) && \text{multiply by } 1 = \frac{P(\tau;\theta)}{P(\tau;\theta)} && (2.31)\\
&= \sum_{\tau} P(\tau;\theta) \frac{\nabla_\theta P(\tau;\theta)}{P(\tau;\theta)} R(\tau) && \text{rearrange} && (2.32)\\
&= \sum_{\tau} P(\tau;\theta) \nabla_\theta \log P(\tau;\theta) R(\tau) && \text{“log-derivative trick”: } \frac{\nabla_\theta P(\tau;\theta)}{P(\tau;\theta)} = \nabla_\theta \log P(\tau;\theta) && (2.33)\\
&= \mathop{\mathbb{E}}_{\tau \sim P(\tau;\theta)} [\nabla_\theta \log P(\tau;\theta) R(\tau)] && \text{return to expectation form} && (2.34)
\end{aligned}$$

An immediate implication is that the gradient is reduced to an expectation that we can estimate it with a sample mean. If we collect a set of trajectories $\mathcal{D} = \{\tau^{(i)}\}_{i=1,\ldots,N}$ where each trajectory is obtained by letting the agent act in the environment using the policy π_θ, the policy gradient can be estimated with

$$\nabla_\theta U(\theta) \approx \hat{g} = \frac{1}{N} \sum_{i=1}^{N} \nabla_\theta \log P\left(\tau^{(i)};\theta\right) R\left(\tau^{(i)}\right). \tag{2.35}$$

Looking at (2.35), we see a dependency on the dynamics model, but there is a way to decompose the path into states and actions, as follows:

[13] We are assuming the discrete action space should be replaced by an integral in the case of continuous action space expectation.

$$\nabla_\theta \log P\left(\tau^{(i)};\theta\right) = \nabla_\theta \log \left[\prod_{t=1}^{H} \underbrace{P\left(s_{t+1}^{(i)} \mid s_t^{(i)}, a_t^{(i)}\right)}_{\text{dynamics model}} \cdot \underbrace{\pi_\theta\left(a_t^{(i)} \mid s_t^{(i)}\right)}_{\text{policy}}\right] \tag{2.36}$$

$$= \nabla_\theta \left[\sum_{t=1}^{H} \log P\left(s_{t+1}^{(i)} \mid s_t^{(i)}, a_t^{(i)}\right) + \sum_{t=1}^{H} \log \pi_\theta\left(a_t^{(i)} \mid s_t^{(i)}\right)\right] \tag{2.37}$$

$$= \nabla_\theta \sum_{t=1}^{H} \log P\left(s_{t+1}^{(i)} \mid s_t^{(i)}, a_t^{(i)}\right) + \nabla_\theta \sum_{t=1}^{H} \log \pi_\theta\left(a_t^{(i)} \mid s_t^{(i)}\right) \tag{2.38}$$

$$= \sum_{t=1}^{H} \nabla_\theta \log P\left(s_{t+1}^{(i)} \mid s_t^{(i)}, a_t^{(i)}\right) \xrightarrow{\quad} 0 + \sum_{t=1}^{H} \nabla_\theta \log \pi_\theta\left(a_t^{(i)} \mid s_t^{(i)}\right) \tag{2.39}$$

$$= \sum_{t=1}^{H} \nabla_\theta \log \pi_\theta\left(a_t^{(i)} \mid s_t^{(i)}\right). \tag{2.40}$$

The last deduction and (2.26) allows us to write the following expression that provides us with an unbiased[14] estimate of the gradient, and we can compute it without access to the dynamics model (model-free method),

$$\nabla_\theta U(\theta) \approx \hat{g} = \frac{1}{N} \sum_{i=1}^{N} \left(\sum_{t=1}^{H} \nabla_\theta \log \pi_\theta\left(a_t^{(i)} \mid s_t^{(i)}\right)\right) \left(\sum_{t=1}^{H} R\left(s_t^{(i)}, a_t^{(i)}\right)\right). \tag{2.41}$$

To determine the final expression, we want to leave two last thoughts: the concept of causality and reward discount. The first has to do with the fact that future actions should not change past decisions. Furthermore, the present actions only impact the future. Therefore, we can change our objective function to reflect this. Finally, adding the discount factor to the expression can reduce variance, reducing distant actions' impact. With these two final notes, our estimation of the gradient becomes

$$\nabla_\theta U(\theta) \approx \hat{g} = \frac{1}{N} \sum_{i=1}^{N} \left(\sum_{t=1}^{H} \nabla_\theta \log \pi_\theta\left(a_t^{(i)} \mid s_t^{(i)}\right)\right) \left(\sum_{t'=t}^{H} \gamma^{t'-t} R\left(s_{t'}^{(i)}, a_{t'}^{(i)}\right)\right). \tag{2.42}$$

The first term of (2.42) is the maximum log-likelihood. In deep learning methods, it measures the likelihood of the observed data. In our context, and like we said before, it measures how likely the trajectory is under the current policy. By multiplying it with the rewards, we intend that the highest reward trajectories are more likely to happen and those that produce the lowest rewards least likely to happen. In short, keep what is working and throw out what is not. Figure 2.3 tries to make this intuition visual.

[14] We know by the law of large numbers that $\frac{1}{N}\sum_{i=1}^{N} \nabla_\theta \log P\left(\tau^{(i)};\theta\right) R\left(\tau^{(i)}\right) \to \nabla_\theta U(\theta)$ with probability one as $N \to \infty$.

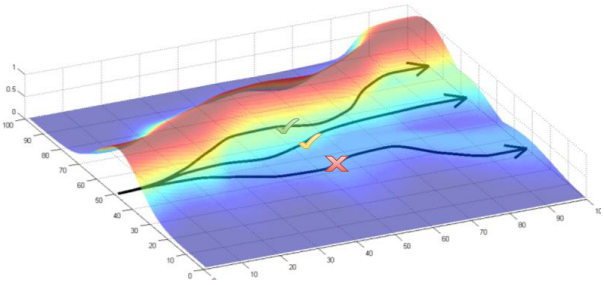


Figure 2.3: Visual intuition of policy gradient using hill trajectories. Source: [40].

If going up the hill means higher rewards, we will change the model parameters (policy) to increase the likelihood of higher trajectories.

2.8 Reinforcement Learning in Financial Markets

After introducing some of the most fundamental RL concepts and methods, it is time to research RL's work in financial markets. Following RL's success demonstrated in [16, 17], many researchers tried to apply these algorithms to the financial trading system.

Wang et al. [41] proposed a deep Q-learning algorithm to build an end-to-end algorithmic trading system. The system can automatically determine what position to hold at each trading time. The database used in the experiment involves 15 years of daily data of the Hang Seng Index and S&P 500 from 2001 to 2015. They set a delta price using data from the past 200 days, had three discrete action spaces (buy, hold and sell), and used long-term profit as a reward. The experimental results show that their proposed system outperformed the buy-and-hold strategy and the strategy learned by recurrent reinforcement learning. Jeong and Kim [42] proposed a very versatile study where the authors combined the Q-learning algorithm with a deep neural network to solve three different problems: predict the number of shares to trade, increase profits in a confused market, and prevent overfitting from insufficient financial data (in the latter case, they used transfer learning). Kim and Kim [11] proposed a framework to optimise trade and stop-loss boundaries using deep Q-network on pairs trading strategy. The agent is trained to select the optimum level of both boundaries, given a spread to maximise the expected sum of discounted future profits. The chosen data was the daily adjusted price of 50 stocks (S&P 500) from 1990 to 2018. In addition to having shown the proposed model is trained well and outperforms traditional pairs-trading strategies, they also studied the optimum window size and compared the trading signal's performance using different linear regression models. Fallahpour et al. [10] employed an N-armed bandit problem to optimise pairs trading parameters like formation and trade period duration, and trade and stop-loss boundaries. The authors took the spread using an error-correction model and found the parameters using a grid-search algorithm. They compared their proposed model with a constant parameter model similar to a traditional pairs-trading strategy. They used intraday one-minute data of U.S. equity market from June 2015 to January 2016. The performance achieved better results when compared to the constant parameter model.

Once again and similarly to what was done for the pairs trading stages before, table 2.3 provides the

RL literature related to financial markets with remarkable impact.

Table 2.3: Literature exploring the application of reinforcement learning in financial markets.

Article	Sample	Contribution
Fallahpour et al. [10]	U.S. S&P 500 2015-2016	Employing of RL (modelling based on the N-Arm Bandit problem) to optimise the formation and trading window, the trading thresholds, and stop-loss
Kim and Kim [11]	U.S. S&P 500 2009-2018	Introduction of a innovative approach to set a dynamic boundary based on a spread in each trading window using DQN
Wang et al. [41]	(i) HK HSI (ii) U.S. S&P 500 2001-2015	Presentation of a deep Q-trading method that can detect market status from raw and noisy data, and pays attention to long-term returns
Jeong and Kim [42]	(i) U.S.S&P500 (ii) HK HSI (iii) EuroStoxx50 (iv) KR KOSPI 1987-2017	Very versatile study combining Q-learning with deep neural networks

2.9 Conclusion

In this chapter, we surveyed the main research topics related to pairs trading and reinforcement learning. First, we analysed and compared the most common approaches for pairs selection, namely the distance approach and the cointegration approach. Next, we explored the standard threshold-based trading model and briefly inspected other trading models suggested in the literature. Finally, we presented the most relevant concepts and methods of RL, and their application in the financial markets.

Chapter 3

Proposed Model

The previous chapter introduced the pairs trading investment scheme and its fundamental notions and the concepts and intuition behind reinforcement learning. In this chapter, the proposed model aiming to optimise the pairs trading strategy is exposed.

3.1 Model Architecture

Figure 3.1 presents a simplified block diagram of the proposed model architecture, and from now on, we will go into more detail about each component during this work. This chapter will start by presenting the alternative method for the trading signal extraction, then moving through the set of rules chosen for selecting pairs and concluding with a detailed exposition of our DRL model. We decided to leave the formalities about the rolling window block to the next chapter because it is a more practical consideration.

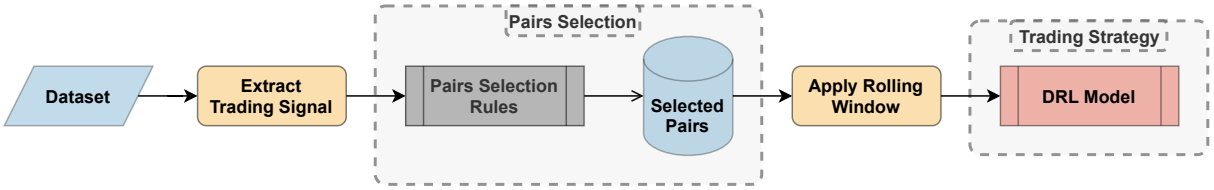


Figure 3.1: Design of proposed model.

The model's flow in a simplistic way can be represented just like figure 3.1. Initially, after the dataset (the chosen securities are ETFs - a detailed description is present in section 4.1) is processed, we extract the trading signals of various possible pairs combinations. Each pair is then subject to a set of rules to determine whether it meets the conditions to be selected (formation period). If the pair is selected, a rolling window is applied to the trading signal, and we reach the last step, the trading strategy. For the trading strategy, we propose the application of a DRL algorithm.

As we had already mentioned in section 1.3, this work is divided into two major study phases represented by the blocks in yellow and red. In section 4.2, we will present the implementation proposal for each study phase. In this chapter, the focus is on justifying and explaining what is behind each of the blocks.

3.2 Trading Signal

There is a fundamental procedure for both stages in pairs trading strategy that in most of the literature, it is neglected, that it is the extraction of the signal for trading. In this section, we will study the possible implications of different approaches to extract the trading signal.

3.2.1 Problem Statement

In the Engle-Granger test, when we first set up the cointegration regression between the price series, Engle and Granger [20] suggested using ordinary least squares (OLS) to determine the cointegration factor (named hedge ratio in pairs trading). This approach seems to be the workhorse among all the literature on cointegration-based pairs trading. However, there are a few problems with OLS, cointegration and pairs trading.

The OLS regression yields predictions of a dependent variable contingent on an independent variable and minimises the sum of squared errors of prediction. It assumes that the independent variable is an observed score known without error, and all of the error is connected to the dependent variable. OLS is not designed to estimate underlying functional relationships, and if both variables contain error, OLS regression tends to yield biased estimates of such functional (or true score) relationships.

Assume that x_i, y_i, and ε_i are an independent variable, a dependent variable, and an error term. The resulting regression equation has the usual form of

$$y_i = \beta x_i + \varepsilon_i. \tag{3.1}$$

The squared discrepancies to be minimised are

$$\sum_{i=1}^{n} (y_i - \beta x_i)^2, \tag{3.2}$$

and we can estimate β from the previous equation by taking a partial derivative,

$$\beta = \left(\sum_{i=1}^{n} x_i' \, x_i \right)^{-1} \sum_{i=1}^{n} x_i' \, y_i. \tag{3.3}$$

It is trivial to see that if we have modelled the regression (3.3) in a different way with the independent and dependent variable switched, the resulting hedge ratio (β_1) will not be the inverse of the other (β_0), i.e., $\beta_0 \neq \frac{1}{\beta_1}$.

We can conclude that the OLS fit is not symmetrical, meaning the resulting hedge ratio is not symmetrical. From a trading standpoint, we want our algorithm to treat the two securities symmetrically. We do not want to ignore the variance (volatility) of one security favouring the other. Furthermore, if the investor reverses its position, the reverse position should be the exact inverse of the original position. The unsymmetrical coefficients imply that a hedge of long on X and short on Y is not the opposite of long on Y and short on X. In short, the hedge ratios are inconsistent.

3.2.2 Total Least Squares

We propose to study a possible better approach of using orthogonal regression along with [43, 44], also referred to as total least squares (TLS) and errors-in-variable (EIV) regression. TLS assume that both variables contain error and seek to identify the line that minimises squared deviations of the data points from the line in both directions. This way, the value of β is calculated consistently.

In the TLS method, both variables are allowed to contain substantial errors. The observed values of Y_i and X_i have the following error terms,

$$Y_i = y_i + e_i, \tag{3.4}$$

and

$$X_i = x_i + u_i, \tag{3.5}$$

where y_i and x_i are the true scores for Y_i and X_i, respectively, and e_i and u_i are random errors that are uncorrelated with each other and with their respective true scores and that have means of 0. Because the errors are uncorrelated with the true scores, the variances in Y and X can be represented as

$$s_Y^2 = s_y^2 + s_e^2 \tag{3.6}$$

and

$$s_X^2 = s_x^2 + s_u^2. \tag{3.7}$$

The error variance ratio is given by

$$\delta = \frac{s_e^2}{s_u^2}, \tag{3.8}$$

and the value of this ratio is taken to be known, at least approximately. Estimating the ratio of the error variances (δ) is generally challenging, but in the example of our study, δ can be unambiguously specified. By including only the measurement errors in X and Y in the TLS model, we are implicitly assuming that equation errors have no role in the analysis. It is assumed that the true score combination between X and Y is linear,

$$y_i = \beta_0 + \beta_1 x_i. \tag{3.9}$$

Under these assumptions, a general maximum-likelihood solution for TLS method yields the following estimation for the hedge ratio,

$$\beta_1 = \frac{s_Y^2 - \delta s_X^2 + \left[\left(s_Y^2 - \delta s_X^2\right)^2 + 4\delta s_{YX}^2\right]^{\frac{1}{2}}}{2s_{YX}}, \tag{3.10}$$

where s_{YX} is the covariance between Y and X.

OLS and TLS can be seen as addressing different questions and serving different purposes. If one wants to predict one variable from another variable, OLS regression is an optimal approach, and TLS is less efficient; in examining the functional relationship between two variables, TLS provides a more plausible model. To conclude, figure 3.2 provides a more visual intuition on both methods for better understanding.

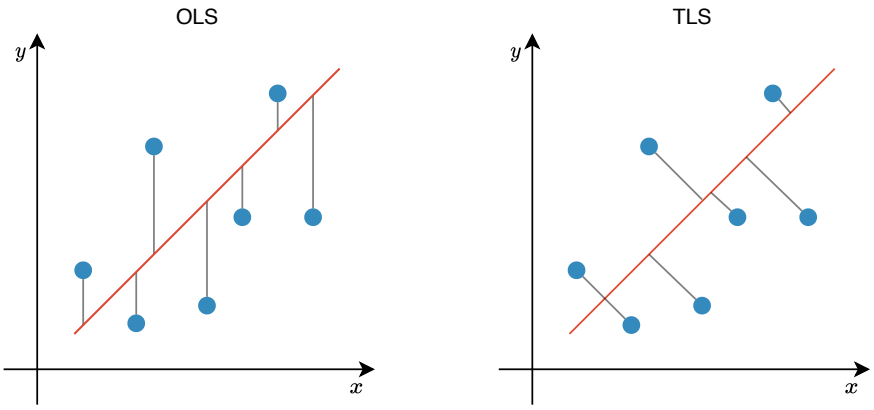


Figure 3.2: Visual comparison between OLS and TLS.

3.2.3 Trading Signal Definition

To sum things up, cointegration analysis using OLS will be sensitive to the ordering of variables. Armstrong [45] noted that the choice of the dependent variable in the Engle-Granger test could lead to different conclusions. One of the relationships may be cointegrated, while the other will not. This could be troublesome because we would expect that both choices will yield the same conclusion if the variables are truly cointegrated.

In practical terms, the hedge ratios obtained by OLS and TLS usually will not differ a lot, but when they do differ, that difference is likely to be significant. So it is worth including the TLS approach in our analysis. To mitigate OLS's issue, we propose to run the Engle-Granger test for both possibilities of choosing the dependent variable and the combination that generates the lowest t-statistic is selected.

In both cases, the value obtained for β is used to decide how we will define buy/sell orders (more details in section 4.3.1). The epsilon value (residual cointegration) is used as a trading signal through Z-scoring, in the state defined based on the formation window.

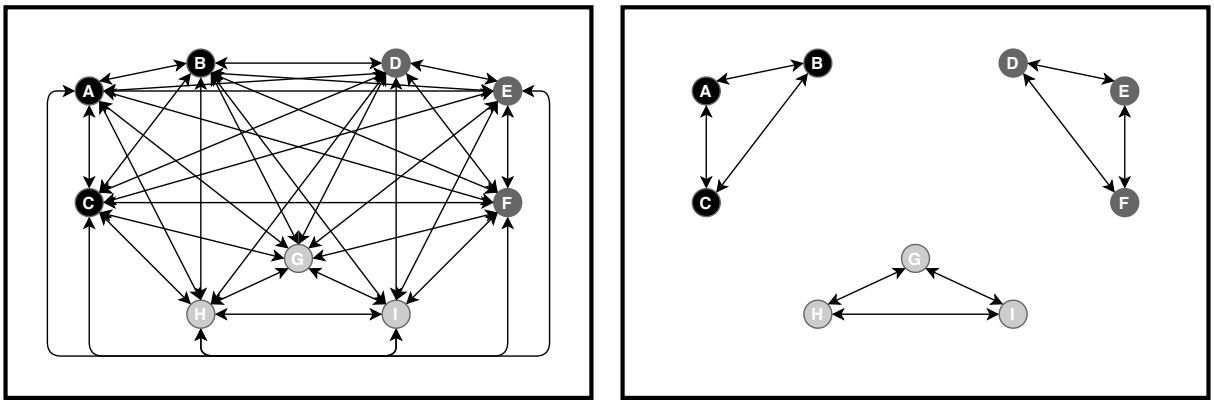
3.3 Pair Selection Framework

This section will focus on the framework developed for selecting pairs to make it as robust as possible, discarding all possible spurious combinations of pairs.

3.3.1 Problem Statement

With the growing popularity of pairs trading, it has become increasingly difficult to find rewarding pairs and their profitability has been declining. In section 2.1, we have already seen the main approaches present in the literature for pairs selection, and this is the stage of the strategy that attracts the most attention from researchers. Although our study's primary focus is on optimising the trading strategy itself, defining a robust pairs selection framework is essential. With that said, there are two vital steps to be defined: (i) the definition of the search space and (ii) the selection criteria of the most promising pairs.

Regarding point (i), the investor starts by selecting the securities of interest (e.g., stocks, ETFs, derivatives, futures) and from there start searching for possible pairs. Here, two methodologies can be found in the literature: an exhaustive search for all possible combinations of the selected securities, without any restriction; or group by sector (or another category of choice), thus restricting the search to pairs within the same sector. Both have already been successfully applied, with the former managing to identify more interesting unusual pairs, while the latter reduces the probability of facing spurious relations. For example, as [46, 47], some research work did not restrict the universe from selecting the pairs. In contrast, [4, 7, 31] organised the securities into categories and selected pairs within those same categories. Figure 3.3 illustrates the two techniques mentioned, with the capital letters identifying the different securities present and the colours representing three hypothetical categories.



(a) Search space without any restrictions.

(b) Search space grouped by category.

Figure 3.3: Search space definition.

The most straightforward procedure is to generate all possible combinations of pairs in the dataset, thus giving us the possibility to find better and possibly less traded pairs. The total number of possible combinations is given by $\frac{n\times(n-1)}{2}$, where n is the number of available securities. The first problem that comes to mind is the computational cost that grows with the number of selected securities. However, a frequent problem also emerges when performing multiple hypothesis tests, known as the multiple comparison problem. This problem introduces a bias that is simply the fact that there is an increased chance to classify those tests incorrectly when running many tests. More specifically, if 100 hypothesis tests are run on random data, with a confidence level of 5%, we should expect to see a false positive rate of 5%. For the reader to understand how relevant this can be to our problem, we will consider

the hypothetical situation in which 20 securities are being handled. There will be $\frac{20\times19}{2} = 190$ possible combinations, which means that approximately nine results will be wrong (5% of 190). Since finding real cointegrated pairs in a randomly selected securities group is quite rare, this number can have much impact. Formally, if m independent comparisons are performed, the family-wise error rate (FWER)[1], is given by

$$\bar{\alpha} = 1 - \left(1 - \alpha_{\text{per comparison}}\right)^m, \tag{3.11}$$

where α represents the probability level of a single test.

This point of view can lead the investor to follow the usual, more restrictive methodology of searching securities only within the same sector. This methodology dramatically reduces the number of statistical tests performed and, consequently, reduces the probability of finding spurious relations. Also, there are fundamental reasons to believe that securities within the same sector are exposed to similar factors. Another advantage of this method is its simplicity of execution in a real trading environment. However, this simplicity of the process can also become a significant drawback. The more investors know these pairs, the more difficult it is to find pairs that are no longer being traded in large volumes, leading to a thinner margin for profits.[2]

Concerning (ii), the investor must define which criteria or set of criteria to select the pairs. When we presented the literature's main approaches, we gave the reader the feeling that the cointegration approach is the superior metric. However, here we expose the multiple comparison problem, an issue with this metric, so it will be essential to overcome this problem.

3.3.2 Search Space

With all this already mentioned in mind, it is time to define our search space. We have already seen that it is a trade-off between:

- Search for pairs without restrictions, resulting in a higher computational cost and leading to the multiple comparison problem, but with the possibility of finding unusual pairs with a higher margin for profits;
- Search for pairs with restrictions by sector, which reduces the number of statistical tests and, in turn, requires a lower computational cost in addition to lowering the multiple comparison problem. However, it has a significant limitation of only allowing us to group pairs in the same sector, making many investors aware of its existence.

Therefore, the commitment made in our study was not to limit the search space. Since the multiple comparison problem cannot be wholly eliminated, its impact can be mitigated if additional verification steps are used.

[1] In statistics, FWER is the probability of making one or more type I errors when performing multiple hypotheses tests. A type I errors is the rejection of a true null hypothesis (also known as a "false positive"). At the same time, a type II error is the non-rejection of a false null hypothesis (also known as a "false negative").

[2] Note that pairs traded in large volumes will not allow for significant divergences from the expected value (mean) and will not allow a margin for profits.

3.3.3 Pairs Selection Criteria

Following the line of reasoning presented above, in our study, we decided to follow the set of criteria presented by Sarmento and Horta [12] to guarantee a more robust pairs selection framework. This set of criteria is a unification of methods applied in separate research works. Hence, for a pair to be selected, the following four conditions must be guaranteed:

1. The pair's constituents are cointegrated.
2. The pair's spread Hurst exponent reveals a mean-reverting character.
3. The pair's spread diverges and converges within convenient periods.
4. The pair's spread reverts to the mean with enough frequency.

We now proceed to describe each step in more detail. As we have seen several times throughout our study, cointegration proves to be the better approach in pair selection, since it identifies econometrically more sound equilibrium relationships compared to other methods (distance or correlation). Therefore, a pair is only eligible for trading if two securities that form a pair are cointegrated. To test this condition, we use the Engle-Granger test due to its simplicity.

Secondly, we have already seen that a cointegrated pair of securities is defined as having a long term stable or stationary relationship but that this not necessarily imply mean-reversion per se. Deviations from equilibrium can occur and be restored throughout time, which implies some mean-reversion properties. The definition of mean-reversion refers to a time series that displays a tendency to revert to its historical mean value. Mathematically, such a (continuous) time series is referred to as an Ornstein-Uhlenbeck process, in contrast to a random walk (Brownian motion), which has no "memory" of where it has been at each particular instance of time.

Ramos-Requena et al. [48] evidenced that the link between market memory and market equilibrium suggests that Hurst Exponent can be an excellent indicator to detect diverging securities and make profitable pairs trading strategies. Therefore, we use the Hurst exponent value as a validation step to have more confidence in the mean-reverting character of the pair's spread, aiming to restrict false positives from the multiple comparison problem. This additional step will guarantee that undesirable data samples made it through the cointegration test, but not mean-reverting can be detected and discarded.

Our criterion is adapted from Ramos-Requena et al. [48], where the authors state that cointegration is an overly restrictive condition and is rarely fulfilled (something Chan [30] also mentions). The authors made a ranking of pairs based on the Hurst exponent and chose the best pairs accordingly. In our study, the Hurst exponent is used as an additional validation step and cannot be seen as a direct replacement of the cointegration test. This is because the Hurst exponent measures the degree of mean-reversion of a time series and not its stationarity. Although in practice, most stationary time series are also mean-reverting, there may be exceptions. As an example, we invite the reader to consider the process X that verifies $X_t = X_{t-1}$ for $t > 0$. Assuming that X_0 takes the value 1 with probability 0.5, and 0 otherwise, we are in the presence of a time series stationary but not mean-reverting, proving that stationarity does not imply mean-reversion.

The Hurst exponent literature has dedicated itself to providing new algorithms for a more efficient Hurst exponent estimation in financial time series. Generalised Hurst Exponent (GHE) is one of the most popular methods for Hurst exponent calculations and was the method used in our study. Considering that our study's focus is not on selecting pairs, if the reader wants to deepen the analytical part of the Hurst exponent value (represented here by H) calculation, we invite to inspect the appendix B.1. In conclusion, a time series can be characterised in the following manner:

- $H < 0.5$ indicates a mean-reverting time series.
- $H = 0.5$ indicates that the time series is a Geometric Brownian Motion.
- $H > 0.5$ indicates a trending time series.

Third, a mean-reverting spread alone does not generate profits. There must be consistency between the mean-reversion duration and the trading period. If, for example, the spread takes an average of 40 days to mean-revert, but the trading period is one month, it will be less likely to find profitable situations. Likewise, a short mean-reversion period is not desirable either. The half-life is defined as the number of periods required for the impulse response to a unit shock to a time series to dissipate by half. It is widely used as a measure of persistence and can quantify the degree of mean-reversion of the deviation. For us, the half-life metric can be interpreted as an estimate of the expected time for the spread's mean-reversion [30], and this step is used to filter pairs whose half-life is not consistent with the trading period.[3]

Finally, we force the spread to cross the mean once a month to provide sufficient liquidity. There is, of course, a (negative) correlation between the number of times the spread crosses the mean and the half-time period, as more mean crosses are naturally associated with a shorter half-life. However, these metrics are not entirely identical since adding this constraint might not merely enforce the previous condition but also discard pairs that do not cross the mean, providing any opportunities to exit a position despite meeting the mean-reversion timing conditions.

Having described each of the steps necessary for a pair to be selected, it is necessary to state the parameters used at each step. Figure 3.4 represents in a flowchart the four steps that each combination of securities must verify in order to be selected for trading:

1. It is imposed for a pair to be considered cointegrated, it must present a p-value of less than 1%.[4]
2. The spread's Hurst exponent must be smaller than 0.5.
3. The half-life period (represented by hl) should lay between one day and one year.
4. It is imposed that the spread crosses the mean at least 12 times per year, which ideally will be equivalent to a minimum of one cross per month, on average.

[3] Like the one above, the analytical calculation of the half-life of mean-reversion is shown in appendix B.2.

[4] As a reminder, in the Engle-Granger test, the null hypothesis used is that no cointegration exists. So we can use the p-value to require the rejection of the hypothesis at a 1% significance level, indicating a good candidate pair.

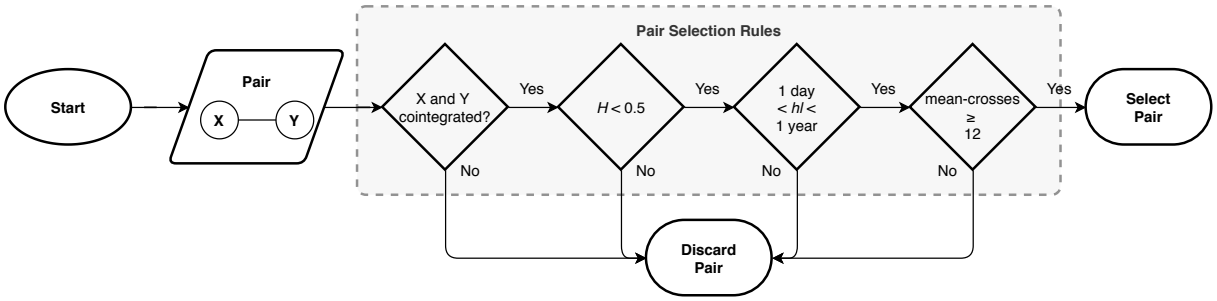


Figure 3.4: Pair Selection Rules. Adapted from: [12].

3.4 Trading Strategy

The first two sections described all the fundamental steps before the trading strategy - the core stage of pairs trading. The definition of the strategy is what leads us to win or lose money. This section will define how we propose to define the trading strategy. It starts with a demonstration of some limitations in the traditional model, goes through the reasons that led us to choose the algorithm used, goes on to the detailed description of the implemented model and ends with some relevant considerations about artificial neural networks.

3.4.1 Limitations and Gaps in the Traditional Trading Strategy

One downside of the traditional threshold-based trading model, described in section 2.2.1, is that the only criterion to set positions is crossing a predefined threshold regardless of the trading signal's current direction. On the one hand, entry points are not precisely defined, leading to periods of remarkable decline in our portfolio if the spread continues to diverge. On the other hand, a position that already had the potential for profits may not be closed, because the spread did not precisely touch the exit threshold, and new divergence may occur. These behaviours can be seen in figure 3.5.

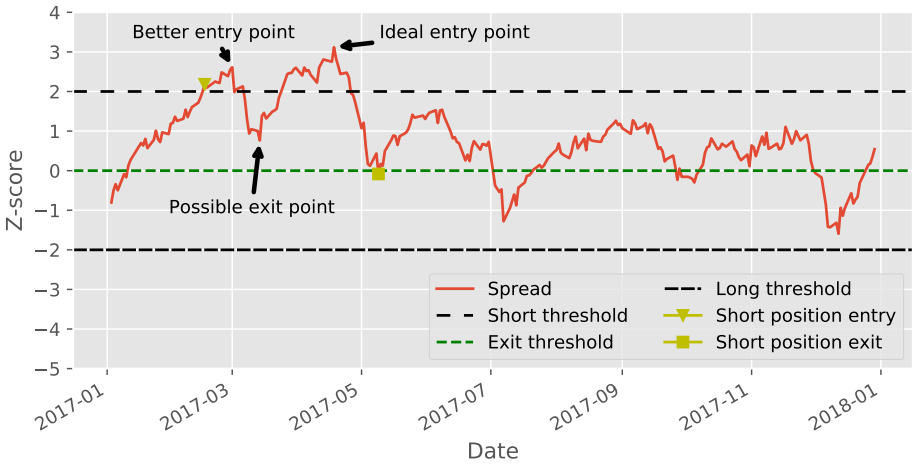


Figure 3.5: Limitations on defining positions in the traditional strategy.

The reader should be familiar with this figure as it is the figure 1.3 with descriptive notes of the previously mentioned considerations. We notice that the spread continues to diverge in the first case,

so losses in the portfolio occur. Gains will only begin to manifest when the signal converges again. In the second case, we have a situation where the position could have been closed, and profits could have been guaranteed. In the third case, the spread diverges again, and we have the ideal entry position. Entering a position at that moment provides more margin for profit as the distance to the mean is higher.

There are still possibilities for profits not covered by this traditional model, which would be to try to predict divergences. If our model detected a period in which the spread tended to deviate from the mean, a position could be opened. Naturally, this is extremely challenging to achieve in practice.

3.4.2 Why Advantage Actor-Critic

These gaps described in the previous section in the traditional threshold-based trading model, motivated us to develop a trading agent based on DRL to maximise long-term profits. This section will describe the reasons and the intuition behind the choice of this algorithm specifically.

Intuition

A particular reason why many supervised learning methods failed in stock market prediction and algorithmic trading is that the market is very volatile. This volatility is not solely caused by short-term noise, e.g., unexpected events, but also by the intrinsic evolution of the market fundamentals, such as economic growth, technology development or social structure. This unpredictability suggests that any models trained with historical data will ultimately fail in new market conditions no matter how accurate they are on the training set. The RL capacity of dealing with new information and adapt to the unknown is absorbing, and that is why we have adopted this form of learning in our work.

First of all, it is time to evoke our current knowledge of RL. As the reader may know, there are two main types of RL methods:

- Value-based: In these methods, we are trying to find or approximate how good it was to reach a particular state or how good it was to take a specific action. The higher the value, the better the state where we are or the action taken. The most famous algorithm is Q-learning and all its enhancements like deep Q-networks (DQN).
- Policy-based: Policy-based algorithms try to find the optimal policy directly without the Q-value as a middleman.

Each of the methods has its advantages. For example, policy-based methods are better for continuous and stochastic environments, while value-based methods are more sample efficient and steady. Most state of the art in RL in financial markets attempts to train a trading agent adopting Q-learning to approximate discounted future rewards, as presented in section 2.8. While Q-learning proves successful in many applications with deterministic discrete actions, its ability to handle stochastic policies is minimal. Due to randomness in financial markets, the optimal trading policy under a particular environment can be stochastic. While policy-based approaches can handle stochastic policies, traditional policy gradient methods using the MC method have high variances. Thus, we choose an actor-critic method (a

middle ground between both approaches), a variation of policy-based RL framework with the potential to output an optimal stochastic policy to train our trading agent, as it allows us to maximise the expected rewards effectively.

Actor-Critic

Actor-critic was created as a combination of the two described approaches. The key idea is to have two models: one for computing an action based on a state and another to produce the action's Q-values.

In our study, we optimise the pairs trading strategy with a type of game using actor-critic. The actor takes as input a state and outputs the best action. It essentially controls how the agent behaves by learning the optimal policy (policy-based). On the other hand, the critic evaluates the action by computing the value function (value-based). Those two independent models participate in this game where they both get better in their role as the time passes. The result is that the overall architecture will learn to play the game more efficiently than the two methods separately. Figure 3.6 represents this new architecture.

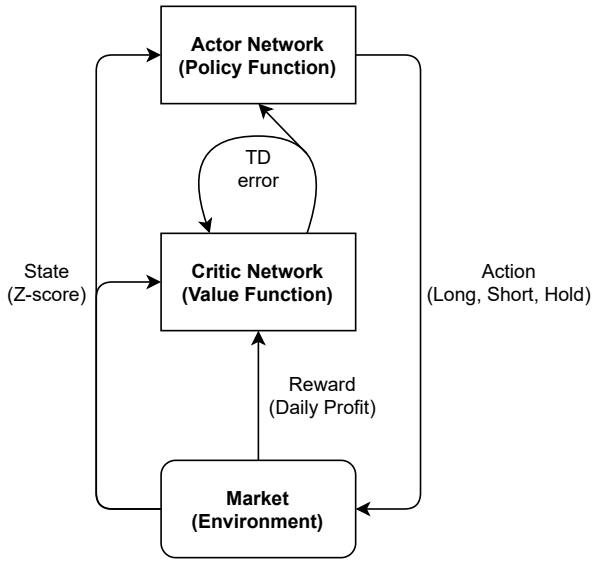


Figure 3.6: The state-action-reward cycle for the actor-critic framework. By observing a state, the policy function (actor) generates an action (long, short, hold) that will be executed in the environment (market). The environment will return the corresponding immediate reward (i.e., the amount of money gained or lost) after the action is taken. The TD error[5] calculated from the reward will serve as feedback to update the value function (critic). The TD error as a critic's estimation for the current policy will act as a signal to update the actor's network. Adapted from: [15].

The actor is a neural network whose task is to produce the best action for a given state. The critic is also another neural network, which receives as input the state of the environment and the actor's action and outputs the action-value (Q-value)[6] for the given pair.

The two networks' training is done separately and uses the gradient descent to update both their weights. At time passes, the actor is learning to produce better actions (starts to learn the policy), and

[5] TD error is part of the update rule in TD learning - (2.23).
[6] Remind the reader that Q-value is essentially the maximum future reward.

the critic is getting better at evaluating those actions. It is relevant to notice that the update of the weights happens at each step (TD learning) and not at the end of the episode, as opposed to policy gradients.

Policy Gradients Improvements

The reader must have noticed that we referred to TD learning previously and the TD error in figure 3.6. However, it should not be self-evident how it comes up. To explain this point, let us first recall the estimation of policy gradient,

$$\nabla_\theta U(\theta) = \mathop{\mathbb{E}}_{\tau\sim\pi_\theta}\left[\sum_{t=1}^{H}\nabla_\theta \log \pi_\theta(a_t \mid s_t)\sum_{t'=t}^{H}\gamma^{t'-t}R(s_{t'}, a_{t'})\right]. \tag{3.12}$$

Traditional policy gradient methods using MC updates (REINFORCE algorithm is the best-known example in the literature) suffer from high variance and low convergence. If we think of a stochastic policy, different episodes can lead us to take different actions. One small turn and can change the outcome entirely. Like we said before in section 2.6.4, MC methods have no bias but high variance. Variance hurts deep learning optimisation. The variance provides conflicting descent directions for the model to learn. One sampled rewards may want to increase the log-likelihood, and another may want to decrease it. Overall, these issues contribute to instability and slow convergence.

To reduce the variance caused by actions, we want to reduce the variance caused by sampled rewards. An obvious solution could be to increase the batch size. However, this solution reduces sample efficiency[7], so we cannot increase it too much. That said, new mechanisms to reduce variance are needed.

Baseline

One way to reduce variance and increase stability is adding a term to the cumulative reward. This addition has no problem as long as the term only depends on the state,[8]

$$\mathop{\mathbb{E}}_{a_t\sim\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a_t \mid s_t)\, b(s_t)\right] = 0. \tag{3.13}$$

This possibility allows us to subtract a term that respects this condition from our expression for the policy gradient, without changing it in expectation:

$$\nabla_\theta U(\theta) = \mathop{\mathbb{E}}_{\tau\sim\pi_\theta}\left[\sum_{t=1}^{H}\nabla_\theta \log \pi_\theta(a_t \mid s_t)\left(\sum_{t'=t}^{H}\gamma^{t'-t}R(s_{t'}, a_{t'}) - b(s_t)\right)\right]. \tag{3.14}$$

Any function b used in this way is called a baseline. Intuitively, making the cumulative reward smaller by subtracting it with a baseline will make smaller gradients and consequently smaller and more stable updates. For the sake of a hypothetical example, let us say that the log-likelihood of three trajectories

[7] Sample efficiency denotes the amount of experience an agent needs to generate in an environment during training to reach a certain performance level.

[8] Suppose that P_θ is a parameterized probability distribution over a random variable, x. Then: $\mathop{\mathbb{E}}_{x\sim P_\theta}\left[\nabla_\theta \log P_\theta(x)\right] = 0$. A proof of this claim can be found in [49].

are $[0.5, 0.2, 0.3]$ and that the exact rewards are $[1000, 1001, 1002]$, respectively. Then the variance of the product of the two terms for these three samples is $\mathrm{Var}\left(0.5 \times 1000, 0.2 \times 1001, 0.3 \times 1002\right)$, which is 23286.8. Now, what if we reduce all reward values by a constant? Assuming a constant of 1000, the variance of the product becomes $\mathrm{Var}\left(0.5 \times 0, 0.2 \times 1, 0.3 \times 2\right)$, which is around 0.93, a much smaller value.

The most common choice of the baseline is the on-policy state-value function $V^{\pi}\left(s_t\right)$.[9] Empirically, the choice $b\left(s_t\right) = V^{\pi}\left(s_t\right)$ has the desirable effect of reducing variance in the sample estimate for the policy gradient. Resulting in faster and more stable policy learning. It is also attractive from a conceptual point of view: it encodes the intuition that if an agent gets what is expected, it should "feel" neutral about it.

We have seen so far that the policy gradient has the general form

$$\nabla_{\theta} U\left(\theta\right) = \underset{\tau \sim \pi_{\theta}}{\mathbb{E}} \left[\sum_{t=1}^{H} \nabla_{\theta} \log \pi_{\theta}\left(a_t \mid s_t\right) \Phi_t \right], \tag{3.15}$$

where Φ_t could be any of the choices seen before. It turns out that there are other valid choices.[10] The on-policy action-value function is also valid, since the definition of $Q^{\pi}\left(s_t, a_t\right)$ is in fact the expected return, starting from state s_t and action a_t, when acting on-policy for the rest of the trajectory. Using $\Phi_t = Q^{\pi}\left(s_t, a_t\right)$ and then using a value function baseline, which we are always free to do, will lead us to what is called in literature the advantage function,

$$A^{\pi}\left(s_t, a_t\right) = Q^{\pi}\left(s_t, a_t\right) - V^{\pi}\left(s_t\right). \tag{3.16}$$

An intuitive way of thinking about the advantage function is to think that it tells us how much better an action is over a random action selected according to the policy π for that state.

In deep learning, we want input features to be zero-centered. RL is interested in knowing if an action performs better than the average. For instance, suppose that two actions for a given state would yield the same expected return. Without the critic, the algorithm would increase the probability of these actions based on the objective function, U. With the critic, it may turn out that there is no advantage and thus no benefit gained in raising the actions' probabilities, and the algorithm would set the gradients to zero.

Nevertheless, there is a problem with the formulation of (3.16). It would mean that we would have to build two neural networks for both the action-value function and the state-value function (in addition to the policy network). This additional computational cost would be very inefficient. Instead, we can use the relationship between the $Q\left(s_t, a_t\right)$ and $V\left(s_t\right)$ from Bellman optimality equation (inspect appendix A.2),

$$Q^{\pi}\left(s_t, a_t\right) = \mathbb{E}_{\pi}\left[R_{t+1} + \gamma V^{\pi}\left(s_{t+1}\right)\right]. \tag{3.17}$$

So, we can rewrite the advantage function as

[9] Recall that this is the average return an agent gets if it starts in state s_t and then acts according to policy π forever.

[10] For a more detailed treatment of this topic, we refer to the work of Schulman et al. [50], which goes into depth about different choices of Φ_t.

$$A^{\pi}(s_t, a_t) = R_{t+1} + \gamma V^{\pi}(s_{t+1}) - V^{\pi}(s_t). \tag{3.18}$$

Rather than having the critic to learn the Q-values, we make him learn the advantage values. That way, the evaluation of an action is based not only on how good it was but on how good it can be. The main advantage is being able to reduce policy networks variance and stabilise the model. A final note regarding the point raised related to TD learning and TD error. How does everything we talk about connect with that? If the reader analyses the advantage function, it is no more than the TD error of (2.23).

Advantage Actor-Critic

The advantage actor-critic has two main variants: the asynchronous advantage actor-critic (A3C) and the advantage actor-critic (A2C). Mnih et al. [51] introduced the A3C algorithm in 2016. In essence, A3C implements parallel training where multiple workers in parallel environments independently update a global network; hence "asynchronous". The key benefit in having asynchronous actors is an effective and efficient exploration of the state space.

A2C is similar but without the asynchronous part; this means a single worker variant of the A3C. OpenAI[11] empirically found that A2C produces comparable performance to A3C while being more efficient. That said, the decision of our study passed through the implementation of A2C.

3.4.3 Deep Reinforcement Learning Model

This study will attempt to implement an optimal pairs trading strategy by taking optimal trading decisions that correspond to the given spread. A pairs trading system can profit if a position is opened and closed in the right times in a trading window. It is therefore considered as a kind of a game. A correct agent decision yields a positive reward, and an incorrect decision yields a negative reward.

Overview

The core of the system is two feedforward neural networks acting as the actor and the critic. The topology of the hidden layers was set empirically, while the output layer is intrinsic to our system's design. The actor output layer comprises three linear neurons, representing the probability of executing each possible action. The critic output layer is composed of one linear neuron.

The agent interacts with a simulated market environment in discrete steps $t = 0, 1, 2, \ldots$. At each of those steps, it receives a state vector s_t as input. After a forward propagation, the actor (policy function π) outputs a probability distribution. For example, the output of π maybe long with probability 0.7, short 0.2 and hold 0.1. We sample an action according to such probability. Since we are dealing with a stochastic policy, our agent can explore the state space without always taking the same action.

[11] OpenAI is an artificial intelligence research laboratory and deployment company.

$A_t \sim \pi(\cdot \mid S_t), \forall t > 0$ is made up of just three action signals due to our discretization of the action space. There is an external imposition on the agent to allow no more than one open position at a time since we assume that all available capital per pair is fully invested when a position is set. So the interpretation of the action depends on whether it is a currently open position, as described in table 3.1.

Table 3.1: Interpretation of action signal.

Action	Current Position	Action Outcome
Long	Long	-
	Short	Close Short
	None	Open Long
Short	Long	Close Long
	Short	-
	None	Open Short
Hold	Long	Hold
	Short	Hold
	None	Nothing

In the column referring to the action's outcome, the two cells without any data mean that we do not allow the agent to take the same action as the current position, making it impossible to secure a double position. For example, if a long position is already open, no new long position can be allowed until the current long position is closed. That said, we have restricted the choice of agent for the two remaining possible actions.

Market Simulation

With the market simulation, we want to create an environment to coordinate the flow of information that reaches the system so that it follows the RL paradigm, which means supplying the system with a state, receiving its response in the form of an action and answering with a new state and a reward (figure 3.6). Also, this process must be consistent with trading in the real stock market, so that the learned behaviour and performance measure would translate to real trading.

The market simulation follows ETFs prices. When it receives a response in the form of an action a_t, it updates all intrinsic information, and drafts a new state s_{t+1} and a scalar reward R_{t+1} for the chosen action. State and rewards signals are produced with functions described next.

The simulated market environment for each pair includes the following information:

- ETFs prices of the pair's constituents;
- Hedge ratio;
- Z-score values;
- Positions history;
- Account balance history.

State

The state consists of the next set of input features for both neural networks. Wang et al. [41] used delta prices to define the states, and the author demonstrated promising results. We have tested this approach empirically, and it has proven to be successful compared to conventional Z-score values. A state is defined as $z_t - z_{t-1}$ of n days in the past, where z correspond to the Z-score values.

Conceptually this approach also makes sense, since we are trying to tell the agent not to focus on the absolute values of the spread, which could lead to a behaviour closer to the traditional strategy, but rather to focus on relative values and try to find opportunities that bridge the gaps in the threshold-based model.

In the literature, we are aware that several works incorporate additional features, such as indexes as a proxy for the state of the market and economy, or market volatility indicators, among others. We stick with this more pure state definition for simplicity and because it is not the goal of our study to test these features' implications.

Reward Signal

The first approach for the reward signal was to use only the profit obtained from each position. This approach is intuitive, simple to implement and imposes no constraint to the system on how to achieve profitability. However, this approach has conceptual flaws, since the agent was not punished for holding positions with negative unrealised profits but was just punished in the moment of their closing. This behaviour led to the agent either being too conservative and not opening any position or learning to keep an open position until its unrealised profit bounced back to positive values no matter how long it took.

It is therefore imperative to consider the variations in unrealised profit, generally referred to as returns. The magnitude of the return defines the reward signal: the difference between the unrealised profit in the state s_t in which the action was taken and the unrealised profit in the state s_{t+1} to which the action lead. Formally, the reward signal is given by

$$R_{t+1} = \frac{A_{t+1} - A_t}{A_t} \times 100, \tag{3.19}$$

where A is the virtual account balance for the pair, and it is the variable which allows us to track the unrealised profit.

The first question that must arise in the reader's mind is whether this is the best reward signal. This immediate reward could be seen as quite noisy, leading to perhaps not so reliable supervision for the model training. Nevertheless, we have tested several variations, such as a reward signal that contains a term linked to long-term profit (e.g., the accumulated wealth over n days) or a more direct approach analysing the trading signal and comparing it to the current position. In all the empirical tests, the daily returns found to be the superior reward signal.

Even so, defining the reward signal only in this way often led the system to get stuck in a local opti-

mum[12], which ended up regularly leading the agent's decisions not to open any position. This behaviour, which was too conservative, motivated us to add a penalty for not opening a position.

Furthermore, some RL systems use risk-adjusted profit as their reward. The most common method to adjust for risk is the Sharpe ratio (this metric will be important in evaluating our study - section 4.4.2), which divides the profit by the standard deviation of unrealised profit. The other alternative is the Sortino ratio, which divides profits by the standard deviation of the unrealised profit downside. Both aim to penalise large variations of unrealised profit which are interpreted as risk. In essence, our reward system is very similar, especially to the Sortino ratio since positive volatility is not punished, as is the case with the Sharpe ratio but rewarded. The difference is that rather than introduce the punishment/reward at the end of the position by adjusting the profit, it is spread out over its life with the return at each step, which possibly alleviates the credit assignment problem (CAP)[13].

A final note pointed out by Henderson et al. [52] is that in gradient-based methods (as used in most DRL), a large and sparse output scale can result in problems regarding saturation and inefficiency in learning. During training, the rewards are standardised to overcome this problem, a technique known as reward clipping or rescaling rewards, which compresses the space of estimated expected returns.

Algorithm

Algorithm 1 summarises all the process of training for one single pair. It takes into account all the components already described.

[12] Although all our efforts to make the system more stable, A2C is still too sensitive and can quickly get stuck in a local optimum.
[13] Recall that the CAP is the problem of determining the actions that lead to a particular outcome.

Algorithm 1 A2C Pairs Trading Algorithm

// Assume parameter vectors θ (actor), θ_v (critic), and episode counter E = 0
Initialize step counter $t \leftarrow 0$
Initialize network weights $\theta \leftarrow 0$ and $\theta_v \leftarrow 0$
Initialize network gradients $d\theta \leftarrow 0$ and $d\theta_v \leftarrow 0$
Get initial state s_t
repeat
  repeat
    Sample a_t according to policy $\pi_\theta(a_t \mid s_t)$ ▷ Long, Short or Hold
    Calculate $V_{\theta_v}(s_t)$
    Perform a_t in market environment
    Receive reward R_{t+1} and new state s_{t+1}
    $y = \begin{cases} R_{t+1} & \text{for terminal } s_{t+1} \\ R_{t+1} + \gamma V_{\theta_v}(s_{t+1}) & \text{for non-terminal } s_{t+1} \end{cases}$ ▷ Bootstrapping
    Store $V_{\theta_v}(s_t)$ and y
    $s_t = s_{t+1}$
    $t \leftarrow t + 1$
  until terminal s_t
  Compute batch loss function in wrt θ and θ_v ▷ Details in section 3.4.4
  Perform batch gradients updates of θ and θ_v ▷ Equation (2.24)
  $E \leftarrow E + 1$
until $E > E_{\text{max}}$

3.4.4 Artificial Neural Networks

The way of defining artificial neural networks for A2C can be found in the literature basically in two prime forms:

1. One neural network for both the actor and the critic functions, i.e. one neural network with two output layers, one for the state value and another one for the action probabilities;
2. Two separate neural networks (one for the actor and the other for the critic).

Without literature consensus and an acceptable form of evaluation, we found empirically that separate neural networks learned more quickly. Both networks are trained separately, but the architecture is the same, changing only the output layer.

The number of neurons in the input layer is defined by our state representation elements, which results in 48 input neurons. Two hidden layers follow this input layer with 128 ReLU neurons each. The Rectified Linear Unit (ReLU) activation function was chosen due to its documented superiority for training multi-layer neural networks [53].

Loss Function

It is essential to note that although we characterise the "loss function" as a loss function, it is not a loss function in the typical sense from supervised learning. When compared to standard loss functions,

there are two main differences:

1. The data distribution depends on the parameters. A loss function is commonly defined on a fixed data distribution independent of the parameters we aim to optimise. Not so true in RL, where the data must be sampled on the most recent policy.

2. It does not measure performance. A loss function usually evaluates the performance metric of choice. Here, we care about expected return, $U(\theta)$, but our "loss" function does not approximate this at all, even in expectation. This "loss" function is only useful because, when evaluated at the current parameters, with data generated by the current parameters, it has the negative gradient of performance.

To summarise, after the first step of gradient descent, there is no more connection to performance. This means that minimising this "loss" function has no guarantee of improving expected return for a given batch of data. We can send this loss to $-\infty$ and policy performance could be disastrous; in fact, it usually will. Sometimes, this outcome might be described as the policy "overfitting" to a batch of data. This should not be taken in a literal sense because it does not refer to generalisation error.

We felt necessary to raise this point because it is frequent for ML practitioners to interpret a loss function as a helpful signal during training (loss going down, training going well). In RL, this intuition is untrue, and we should only care about the average return. The loss function means nothing.

Actor Loss Function

We formulate the actor loss based on policy gradients with the advantage function and compute single-sample (per-episode) estimates,

$$L_{actor} = -\sum_{t=1}^{T} \log \pi_\theta \left(a_t \mid s_t\right) A^{\pi}_{\theta_v} \left(s_t, a_t\right), \tag{3.20}$$

where T is the number of timesteps per episode (which can vary per episode), s_t is the state at timestep t, a_t chosen action at timestep t given state s, π_θ is the policy (actor) parameterized by θ, $A^{\pi}_{\theta_v}$ is the advantage function based on the value function (critic), $V^{\pi}_{\theta_v}$, parameterized by θ_v. We add a negative term to the sum since we want to maximise the probabilities of actions yielding higher rewards by minimising the combined loss.

Optimising the loss function with (3.20) could result in converging too quickly to a sub-optimal solution, i.e., the probability of a single action is significantly higher than any other, causing it always to be chosen. To prevent this of happening, we add a penalty based on the entropy of the policy. The entropy used is the Shannon entropy[14], which corresponds to the spread of action probabilities. If the policy outputs actions with relatively similar probabilities, then entropy will be high, but if the policy suggests a single action with a higher probability, then entropy will be small. We use the entropy to improve exploration by encouraging the model to be conservative regarding its sureness of the correct action. We

[14]Defining the self-information of an event $X = x$ to be $I(x) = -\log P(x)$. Shannon entropy quantifies the amount of uncertainty in an entire probability distribution, $H(x) = \mathop{\mathbb{E}}_{x \sim P}[I(x)] = -\mathop{\mathbb{E}}_{x \sim P}[\log P(x)]$.

multiply the entropy by a small constant (0.01) to prevent the penalty from being too high and preventing any convergence at all.

Critic Loss Function

The critic loss function is simpler, being nothing more than the difference of our estimated return using the "TD target" ($R_{t+1} + \gamma V(s_{t+1})$) and the value function $V(s_t)$. So, training the critic can be set up as a regression problem with the following loss function:

$$L_{critic} = L_\delta \left(R_{t+1} + \gamma V^\pi_{\theta_v}(s_{t+1}),\ V^\pi_{\theta_v}(s_t) \right), \tag{3.21}$$

where L_δ is the Huber loss[15], which is less sensitive to outliers in data than squared-error loss.

Hyperparameter Optimisation

One significant disadvantage pointed out to systems involving deep learning models is the complicated process of optimising the network's configuration. Tuning hyperparameters can have a tremendous influence on the results, but at the same time, it is very time-consuming. Since the optimal values depend on the data and the problem, little is known about evaluating or selecting the correct hyperparameter values. It is also empirically and theoretically proven that trials are more efficient than the computationally expensive exhaustive grid search [54]. Thus, a compromise must be achieved. Due to the limited computational resources, the feedforward neural networks' tuning is constrained to the most relevant variables (number of inputs, number of hidden layers, nodes in each hidden layer and learning rate) and their variables set using a trial-and-error approach. The algorithms are run in different settings, and the best-observed results are chosen.

Regularisation Techniques

Artificial neural networks are impressive machine learning systems, so impressive that they suffer from overfitting. Regularisation is the set of techniques used to lower the complexity of a neural network model during training to prevent overfitting. We proceed to describe two widespread and efficient regularisation techniques adopted in this work, L2 regularisation and dropout.

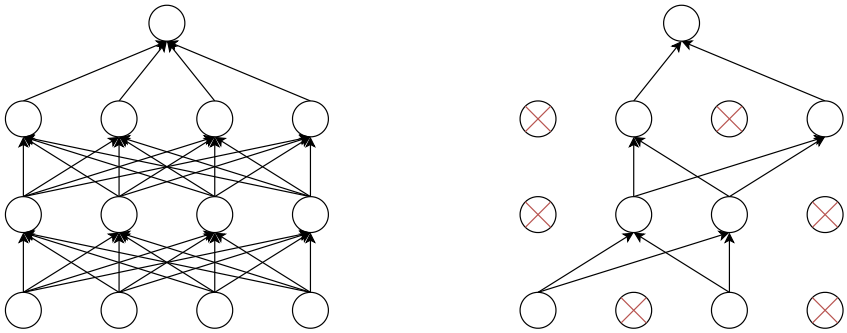
In short, performing L2 regularisation (also commonly known as ridge regression) encourages the weight values towards zero (but not exactly zero). Intuitively, smaller weights will reduce the impact of the hidden neurons. In that case, those hidden neurons become neglectable, and the overall complexity of the neural network gets reduced. Less complex models typically avoid modelling noise in the data, and therefore, tend to reduce overfitting.

The best way to get around overfitting would be to combine the predictions of many different neural networks. However, neural networks' powerful learning ability is usually a prolonged process (especially

[15] The Huber loss is a combination of the mean squared error (MSE) and the mean absolute error (MAE). So, the function is quadratic for small values of the residuals (difference between the observed and predicted values), and linear for large values. It can be expressed as $L_\delta = (y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{for } |y - f(x)| \leq \delta, \\ \delta |y - f(x)| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases}$

when dealing with a large number of parameters), which makes this procedure impractical. Dropout has emerged as the most practical solution to approach this reality. It approximates the training of several neural networks with different architectures while training a single network. This effect is achieved by dropping units (i.e. neurons) - along with their connections - from the neural network during training. This prevents co-adaptation amongst each neuron, trying to extract the individual power of each neuron. Simultaneously, it emulates the training of different network configurations, bringing us closer to the effect of averaging the predictions of multiple networks.

Figure 3.7 shows on the left an example of a standard neuronal network, and on the right, a possible random configuration found in training.



(a) Standard neural network. (b) Neural network with dropout.

Figure 3.7: Dropout application in a small neural network model.

3.5 Conclusion

This chapter started by justifying why we propose an alternative method to extract the trading signal. We then describe the framework adopted for pairs selection. Afterwards, we emphasised the limitations in the traditional trading model. Motivated by these flaws, we presented our proposed model based on DRL and the justification behind our choices. Finally, we conferred some more technical details regarding artificial neural networks.

Chapter 4

Test Planning and Validation

So far, we have discussed the state of the art of pairs trading plus the whole background of reinforcement learning (chapter 2) and detailed all aspects of the proposed model (chapter 3). In this chapter, we will present the dataset used and all data processing conducted. Besides, we will raise the most critical considerations in the implementation of the two suggested study phases. Finally, the characteristics of the trading simulation and evaluation metrics used are also highlighted.

4.1 Dataset

In this section, we will illustrate the dataset chosen to conduct our experiments. We begin by describing the type of securities used, as well as the reason for their use. Then, a more detailed description of the dataset is given. Next, we explain how the data is prepared for the application. Lastly, we define how the data is split into several periods for various trials.

4.1.1 Exchange-traded Funds

Exchange-traded funds (ETFs) are the security of choice in our study. An ETF is a financial instrument that involves a collection of securities (such as stocks, commodities, or bonds) that generally track an underlying index, just like an index fund, but traded on an exchange, just like a stock. ETFs could provide some risk-minimising elements that cannot be obtained when using single stocks in light of the risk associated with pairs trading. To further support, why should be interesting to adopt ETFs as our security of choice, it is essential to introduce the two primary forms of risk to which we are exposed.

There are essentially two dominant risks in pairs trading strategies, horizon and divergence risk [55]. Horizon risk refers to the risk that convergence may not be realised during the intended time horizon (trading period). Divergence risk refers to the risk that the pair continues to diverge after entering a position in the market. It means that the pair will produce negative returns until the end of the trading period or the position is liquidated.

In its nature, an ETF tries to replicate the return on an index consisting of multiple securities, achieving diversification benefits as it is exposed to a basket of assets. As a result, ETFs are not exposed to

idiosyncratic risk[1] to the same extent as single stocks. Price developments of ETFs reflect a market-weighted development of the single stocks within the index. Therefore, changes to one of the stocks' fundamental value do not significantly affect the portfolio composition with ETFs as if it had been a single stock against another single stock in a pair. To conclude, we believe that pairs trading on ETFs reduce both horizon and divergence risks.

In addition to these considerations, ETFs have gained noticeable traction globally with new creations every year. Another exciting point is that many new ETFs differ only marginally from existing ones, leading them naturally to form good potential pairs.

4.1.2 Data Description

In this work, we will focus on a set of ETFs composed only by commodity-linked ETFs. This subset of ETFs is the dataset of choice in the work of Sarmento and Horta [12]. Since we had also adopted the criteria of selecting pairs based on this work, it made sense for us to use the same dataset to allow a greater degree of comparison of this work with existing literature. One reason for restricting this group of ETFs is the reduced number of pairs at the outset, making everything computationally faster and allowing a more careful analysis.

The data used must be available for trading in January 2020. Therefore, corresponding ETFs were selected, leaving a total of 208 ETFs. The universe of commodity-linked ETFs can be distinguished into different categories depending on the tracked commodities. We have chosen to classify them into five different categories following Sarmento and Horta [12]. Table 4.1 describes each of these categories. Appendix C contains a complete list of all eligible ETFs.

Table 4.1: Commodities categories considered in the dataset. Adapted from: [12].

Category	Description
Agriculture	ETFs tracking agricultural commodities, including staple crops and animals produced or raised on farms or plantations.
Base Metals	ETFs chasing energy commodities, which are mostly hard commodities that are mined or extracted. It includes fossil fuels like coal, oil and natural gas.
Broad Market	ETFs following the movement of the commodity market as a whole, rather than a specific sector.
Energy	ETFs monitoring precious metals, naturally occurring metallic elements with high economic value.
Precious Metals	ETFs focused on base metals which support a whole range of industrial and commercial applications including construction and manufacturing.

It is worth mentioning that by choosing ETFs active through an entire period; we are aware that survivorship bias is introduced. Survivorship bias occurs when the performance results use only survivors ETFs at the end of the trading period and excluding those that no longer exist, which means that we will not consider the results of possible positions that had to be left abruptly because of delisted ETFs. To try to limit the impact of this bias, the most recent possible periods were considered.

[1] Idiosyncratic risk refers to the inherent factors that can negatively impact individual (or niche groups) securities. In contrast, systematic risk refers to broader trends that impact the overall financial market.

4.1.3 Data Preparation

Having selected ETFs' universe for our study, we retrieve the price series for each ETF. Here, there is a question that arises, is our data pristine?

Expectations	Reality
• Perfect data recorded minute by minute.	• Data is an amalgamation.
• No gaps or missing data points.	• Not all ETFs trade every day.

We collected the adjusted daily closing prices from January 2, 2009, to December 31, 2019. Those ETFs not trading during an interval will automatically fill with NaN (Not a Number) the non-trading days. So we need to address what to do when we do not have data between two separate dates. We can either interpolate or fill the missing data following a specific approach. The first impulse would probably be to interpolate, and so we would estimate a line between those two dates and then fill in at each point an interpolated value. The problem with this approach is that there was no trading between those two dates; actually, there was no price for that data. So, we can not leave it empty, but we should not interpolate it either. One thing we can do is we fill forward. We go over all the data, and when there is some missing data, we fill forward from the previously known value.

The reason for using this technique instead of the interpolation is the following. Let us suppose we were looking for patterns in the data (precisely the case with our DRL model or even with the threshold-based model) and let us assume we are in a moment given by an interpolated value, what is happening is that we are providing information about the future. For example, we are observing that the price is going up. So if we make any calculation here, we would be inducing a look-ahead bias (peeking into the future), which is not admitted. So we need to stick with only filling forward a last known price. If we do that, then we are not picking into the future.

This solution is ideal because, in addition to solving the problem of incomplete data without tampering the future, it does not fill in data at the beginning, that is, in the case of, for example, an ETF that was not yet traded until a specific date.

Nonetheless, we fill a very conservative number of values, and if after that, an ETF still has some values missing, that ETF is directly discarded. Then, to increase our results' robustness and replicate realistic trading environments as closely as possible, we remove ETFs not verifying a minimum liquidity requisite. This constraint is essential to ensure that the bid-ask spread's transaction costs are consistent[2] (detailed information in section 4.3.3). Following the criterion adopted in [4, 12, 24, 56], we use trading volume to filter out ETFs that have at least one day without trading.

Finally, some price series contain sporadic outliers. There is abundant literature on detecting and treating outliers, including cluster analysis or ML-based techniques. However, these events in our dataset are relatively rare, and therefore a considerably more straightforward methodology has been adopted. For each price series, the return series has been calculated. Then, for each point where the

[2]Very briefly, we will consider a constant bid-ask spread among all ETFs. Trading illiquid ETFs would result in a higher bid-ask spread, which would lead to overestimated results because a higher bid-ask spread would have a significant impact on profit margins.

percentage change is more than 30%[3], it is considered an outlier. Again, as this is a rather unusual situation, each case is treated individually. If a single outlier is detected in the ETF time series, it is manually corrected by analysing other data sources. When more outliers are present, the ETF is discarded.

Figure 4.1 briefly summarises all the stages of data preprocessing. Only after passing through each of these stages will the data be considered ready to start the proposed study.

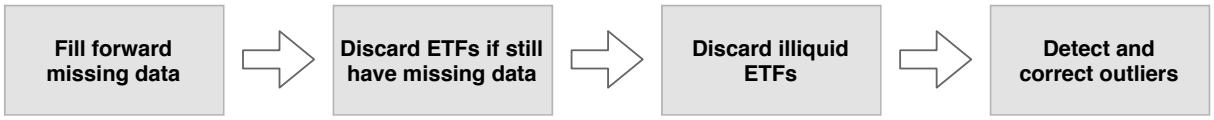


Figure 4.1: Data preprocessing stages.

4.1.4 Data Partition

For each simulation, the data must be partitioned in two periods: formation period and trading period. The formation period simulates the data available to the investor before enrolling in any trade. It is used to find the most appealing candidate pairs. In the case of our DRL model, it is also used to train the model. The trading period simulates how the implemented trading model would perform with future unseen data (also called out-of-sample data).

Reliable results depend on how we expose our solution to the most diverse conditions. By examining several partitions of the dataset, we can gain more confidence in the results' statistical significance. The typical procedure in classical ML problems is to use cross-validation. However, it does not fit financial data (or other intrinsically ordered data - time series in general) applications. The reason is that it can permit peeking into the future (look-ahead bias), and it can lead to unrealistically optimistic results. One way to avoid this problem is with role forward cross-validation. That means our training data is always before our testing data. Nevertheless, we can still have multiple trials just by rolling our data forward.

The periods for simulating each study phase are exhibited in figure 4.2. There are two different configuration possibles, depending on the study phase. In study phase 1, we are going to consider a 2-year long formation period. In section 4.2.1, we will discuss with detail why we chose this length. In study phase 2, a 7-year long formation period is proposed. Here, more formation data is required to fit the DRL model.

[3]This number cannot and must not be too restrictive. Recall that our strategy profits from the divergence of the pair's constituents.

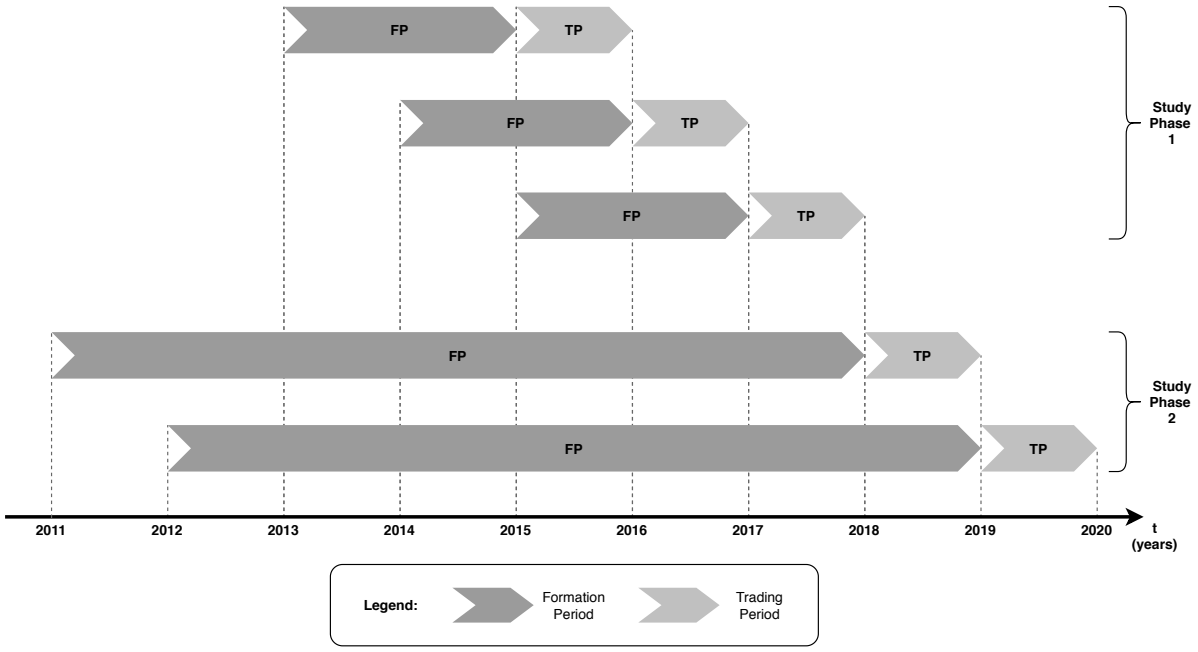


Figure 4.2: Data partition periods.

Table 4.2 presents information that we consider relevant for each of the chosen periods. In periods covering more distant years, the number of ETFs available is considerably lower. This behaviour is natural since the chosen dataset mostly comprises relatively recent ETFs.

Table 4.2: Relevant information regarding data partitioning.

Study Phase	**Study Phase 1**			**Study Phase 2**	
Period	**2013 - 2015**	**2014 - 2016**	**2015 - 2017**	**2011 - 2018**	**2012 - 2019**
Formation Period					
Begin	2013-01-02	2014-01-02	2015-01-02	2011-01-03	2012-01-03
End	2014-12-31	2015-12-31	2016-12-30	2017-12-29	2018-12-31
Trading Period					
Begin	2015-01-02	2016-01-04	2017-01-03	2018-01-02	2019-01-02
End	2015-12-31	2016-12-30	2017-12-29	2018-12-31	2019-12-31
Number of Samples	756	756	755	2012	2012
Formation Period	504	504	504	1761	1760
Trading Period	252	252	251	251	252
Number of ETFs	101	114	120	83	85

4.2 Study Design

Testing the proposed model as figure 3.1 was impossible due to the extreme process of training the DRL model. Dividing the study into two phases is essential for this reason and allows us to study the

impact of each set of blocks separately. The reader should be familiar with this division since it had been mentioned in the definition of the objectives (section 1.3). In this section, we will present, in detail, the design behind each of the study phases.

4.2.1 Study Phase 1

This section will reveal the central technical details of study phase 1, namely the specification of window sizes, and the practical considerations for implementing these windows. We also detail the parameters used in the standard trading model and, finally, we display the general design of this study phase.

Rolling Window approach and Linear Regression analysis

Gatev et al. [4] initially presented a 12-month formation period followed by a 6-month trading period. Huck and Afawubo [26] and Smith and Xu [57] have conducted empirical tests on the lengths of the two periods. Huck and Afawubo [26] consider a formation period of 12 and 24 months respectively and conclude that the 12-month formation period yields the highest average monthly return. In turn, Smith and Xu [57] consider a 9-month formation period versus a 12-month formation period and conclude that a 12-month period provides superior results. In contrast, Kim and Kim [11] tested six different window sizes, concluding that the smallest window achieved the best performance. That is, forming pairs over 30 days and trading them over the next 15 days.

That said, there seems to be no consensus in the literature on the best window size to use, leaving only the idea that the 12-month formation period followed by a 6-month trading period comes out in the lead. The window size decision will depend on the type of strategy and dataset adopted, so it is fundamental to study our dataset and our implementation choices for the ideal window size to use.

The only decision that appears unanimous in the literature is that the formation period's size should be twice the trading period's size. Therefore, in this study, we propose to test six different window size cases, always keeping in common this 2:1 relationship between the formation window and the trading window. Table 4.3 displays the various window sizes considered both in the number of months and in the respective number of days.[4]

[4]Remember once again that on average 1 year = 12 months = 252 days.

Table 4.3: Window sizes considered.

Formation Window		**Trading Window**	
Months	Days	Months	Days
2	42	1	21
4	84	2	42
6	126	3	63
12	252	6	126
18	378	9	189
24	504	12	252

To ensure a fair and unbiased comparison, the number of pairs and the pairs themselves must be the same. Additionally, it is necessary to ensure that the total trading period covers the same period. Before continuing, it is essential to distinguish between the total formation and trading periods, and the formation and trading windows. We request a total 24-month formation period (largest formation window considered in this study) from which the pairs are selected, keeping constant the pairs for all the windows sizes. Consequently, a total 12-month trading period is applied. This structure should be familiar to the reader since it is presented in figure 4.2. By applying a 12-month trading period, a full-year cycle will be taken into account when trading each pair, thus eliminating any potential seasonal biases. To manage shorter windows, we propose to use a rolling window scheme. In this scheme, in each formation window, the values of hedge ratio, mean and standard deviation are recalculated to be applied in calculating the Z-score for the next trading window. Figure 4.3 visually illustrates how this process is done for the example where we consider a 4-month formation window and a 2-month trading window.

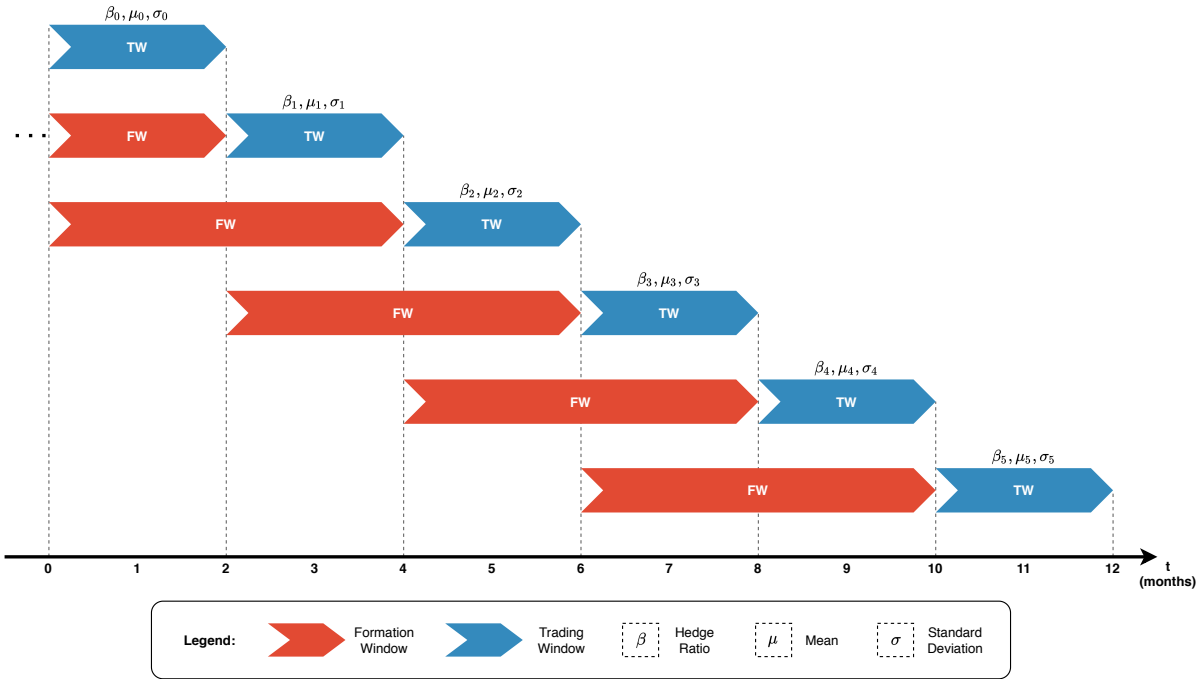


Figure 4.3: Rolling window scheme.

We can see that we obtain six trading windows in total $\left(\frac{12\,\text{months}}{2\,\text{months}} = 6\right)$. In a window transition, if a position is open, it is closed. This closing, although it may generate a positive profit, there is also a possibility that losses will occur. On the other hand, this window transition allows a finer recalibration of the trading signal parameters representing the pair's constituents' current relationship more accurately.

Gatev et al. [4] suggest starting the strategy every month, meaning that a new trading period is initiated every month (with fixed 6-month trading window size). Thus, six overlapping portfolios will always be active at any given time. However, we decided to follow the approach of Papadakis and Wysocki [58] of considering non-overlapping trading windows (which means there is only overlap in the formation window). This approach prevents the overstatement of excess returns due to potential trading on the same pair in several portfolios.

To extract the signal for trading, we opt to analyse the OLS and TLS methods already detailed in section 3.2. It should be noted that we have two separate trading signals that are studied independently. The parameters (β, μ, σ) recalibrated in each window are determined based on the linear regression in question.

Trading Setup

This study phase's primary goal is to compare the impact on the results of the window size used and the linear regression chosen. We are not concerned about optimising the trading model. So, we applied the standard threshold-based trading model described in section 2.2.1 with the parameters specified in table 4.4.

Table 4.4: Threshold-based trading model parameters.

Parameters	Values
Long Threshold	$\mu - 2\sigma$
Short Threshold	$\mu + 2\sigma$
Exit Threshold	μ

These model parameters are suggested by Gatev et al. [4] and are the reference parameters for several studies in this field. The spread's mean, μ, and standard deviation, σ, together with the hedge ratio, β are calculated concerning the formation window under consideration. In practical terms, this μ and σ values will be 0 and 1, respectively, since we work with Z-score values.

Figure 4.4 visually present the full design for this study phase 1.

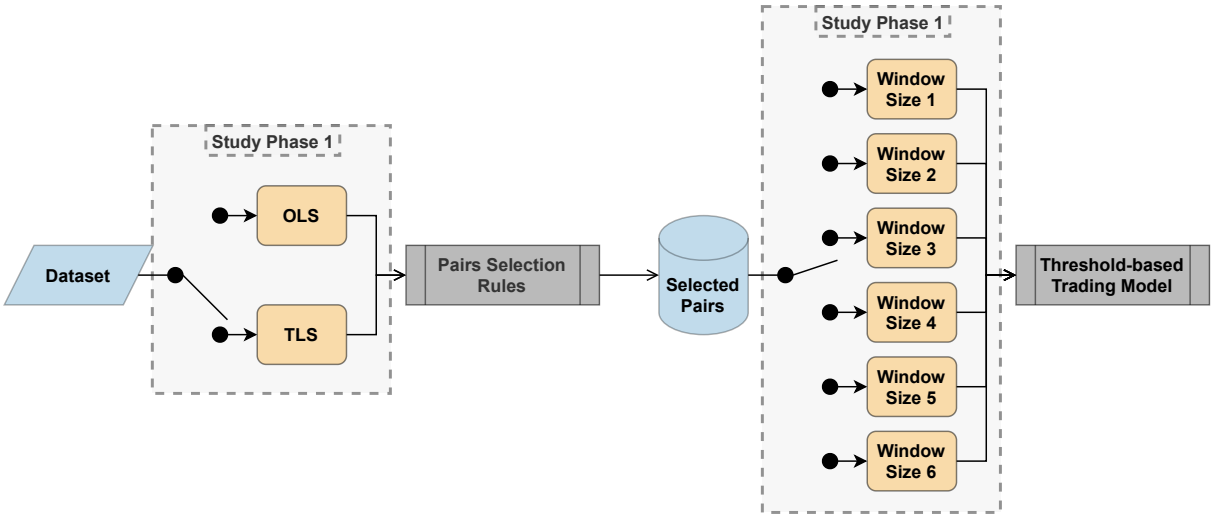


Figure 4.4: Design of study phase 1.

4.2.2 Study Phase 2

This section will highlight some details of implementation to be taken into consideration in the study phase 2. These include some model limitations, how the DRL model interacts with the dataset, and how its implementation is done.

Our DRL model has been built from scratch and carefully developed to ensure a fair comparison with the standard threshold-based model. Due to the limited resources available (computational and time) and also the extraordinary complexity of our model, there are several constraints on the number of experiments we can investigate. There are many hyperparameters to tune, and they are related to each other in complex ways: by changing one of them, a number of the others may no longer be optimal. This tuning is an iterative manual process solely based on feedback from the results obtained. In this way, it is possible, and perhaps most likely, that we are not in the presence of the model with the optimal hyperparameters.

The model interacts with the dataset in two different ways. First, in training, the model's actions have an exploratory component, and the experiences are observed and saved to be used to update the weights of the networks. Secondly, in the test, the model chooses the actions that it believes to be the best and no updating of the networks is done. Naturally, when we mention the dataset passing in the training period, we refer to the part of the dataset stipulated for training (formation period) and the same applies to the test (trading period) - see figure 4.2.

Comparing our DRL model using the six different window sizes proposed in study phase 1 would be ideal but would also be very costly. Therefore, we propose to use the window size that proves to be the most appealing. Similarly, we propose to analyse the linear regression that showed the best results for the two methods to extract the trading signal (OLS and TLS). Regarding pairs selection, using all eligible pairs during the selection stage would be unaffordable. Hence, the pairs were ordered according to the smallest t-statistic as the representative test. The top 10 pairs were elected according to this ranking and used in this second phase of the study.

Equivalent to the previous section, figure 4.5 illustrates the full design for this study phase 2.

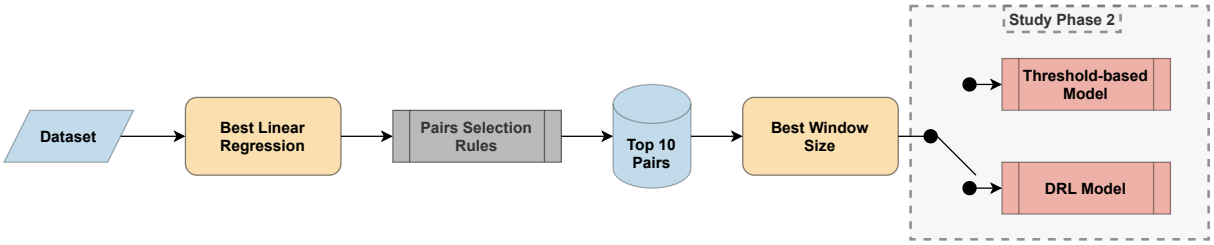


Figure 4.5: Design of study phase 2.

4.3 Trading Simulation

Everything we have presented so far is essential, but we must ensure that our simulation conditions could translate into real trading. This section answers these questions:

1. How to build the portfolio?
2. Why not consider the one-day delay in our simulation?
3. Which transaction costs to take into account?

4.3.1 Portfolio Construction

Depending on the period considered, the portfolios to be studied may vary in size. We have decided to consider that all pairs are equally weighted in the portfolio. With this approach, portfolio returns are calculated by only averaging all pairs' performance, with no need to concern relative proportions of the initial investment. We are aware that assigning weights to each of the pairs in the portfolio can positively impact portfolio returns, and it is a field of many studies in the literature, namely computational intelligence linked to finance. However, we stick to the framework of giving the same weight to all pairs for simplicity.

We still need to define how we are going to allocate the capital for each pair. In theory, if the same value is held in both long and short positions, the capital earned by the short position could be used to cover the long position, and thus, no initial investment would be necessary. This situation is a so-called self-financing strategy [59]. However, in a more realistic scenario, this is not entirely possible. To short a share, an investor needs to deposit an initial investment, known as collateral. According to U.S. regulation, the short-seller has to post 102% of the borrowed amount as collateral for the short position loan. In this study, we assume the collateral value to be the exact value of the ETF being shorted (100%). We believe that this small difference has a marginal impact on the final results, as we are also not considering the interest rate paid by the lender to the investor. Afterwards, to go short a given share, the investor sells the borrowed share and obtains cash in return. From the cash, the investor finances the long position. This behaviour makes it possible for the investor to leveraging its investment. This

leveraging approach is standard behaviour in hedge funds to increase the absolute return in the portfolio, and for that reason is contemplated in this work.

To simplify calculations, we assume an initial investment of one dollar in each pair. This approach is also motivated by being the approach of choice for several authors in the field [2, 4, 12, 24, 31, 46]. Besides, with a one-dollar amount, the return obtained by the pair can be interpreted directly. Here, it is important to reinforce the point raised at the end of section 2.2.1, whereby entering a position with a one-dollar investment, we are necessarily considering that it is possible to buy share fractions. In most cases, this is not true. This problem is quickly overcome if the amount of money invested is increased, allowing us to find a common multiple among the pair's ETFs.

Having decided to invest one dollar in each pair, we still need to decide how to deal with the hedge ratio. We could stick with the approach of Gatev et al. [4] and Rudy et al. [31] of ignoring the hedge ratio to guarantee a dollar-neutral position (investing $\$1$ in the long position and $\$1$ in the short position). However, we prefer to follow the criteria of Sarmento and Horta [12] and Rad et al. [24] and respect the hedge ratio between the pair's constituents. Therefore, we propose to set a maximum of $\$1$ initial investment, ensuring that neither long position nor short position costs more than $\$1$. The way we define the market position follows the framework proposed by Sarmento and Horta [12], and it is represented in figure 4.6.

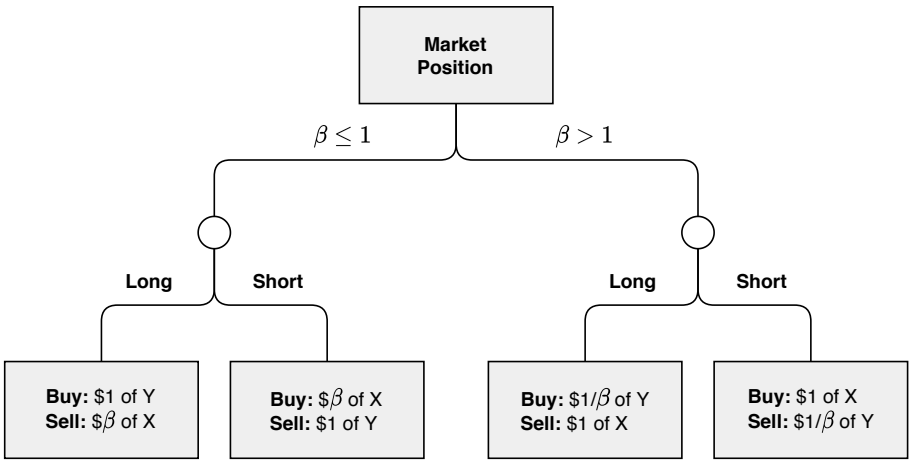


Figure 4.6: Market position definition. Adapted from: [12].

As the trading progresses, we consider that all the capital earned in a trade position is reinvested. For instance, if for example, a first trade has a return of 10%, in the next position being set our initial investment will be $\$1.1$, and not $\$1$. This mechanism makes it easier for us to calculate returns and brings us closer to the behaviour of hedge funds and many investors.

4.3.2 One-day Delay

When opening a position in a pair, several studies [4, 19, 57] propose using a one-day later rule, meaning that setting a position is delayed within one day after deciding to open it. The one-day delay

is an attempt to mitigate the bid-ask bounce[5], which may be a source for an upward bias of the return. Such bias arises because the price of two securities in the pair fluctuates within the boundary of the bid and ask prices. The argument for the one-day delay is the extreme case that the security to short (buy) could be quoted as an ask (bid) quote when the trading signal is triggered. The reason for delaying the trade is the assumption that next day's prices are equally likely to be at the bid or ask price. Gatev et al. [4] interpret the loss of profitability between waiting and not waiting to be equal to the transaction costs. However, this is arguably a noise proxy for transaction costs, as the loss of profitability might also be due to other external factors or potential missed opportunities for opening a position due to rapid convergence. In turn, Chen and Li [19] stated that delaying the trade by one day will reduce the excess return by half of each security's bid-ask spread. So, instead of waiting one day to open a position, we decide to pay the entire bid-ask spread as a transaction cost, which will be covered in section 4.3.3. This choice is also made because waiting one day can also misrepresent our model's decisions' profitability.

4.3.3 Transaction Costs

Strangely, transaction costs are one of the more forgotten elements in pairs trading literature, despite it being a critical factor for a profitable strategy [56]. Pairs trading is associated with numerous transactions throughout the trading periods, making it a more costly strategy than others (e.g., traditional buy and hold strategy). Neglecting transaction costs can lead to overestimated results. We assure that all results presented in this work account for transaction costs. To define transition costs, Do and Faff [56] is the most cited work in pairs trading literature, and outline three transaction costs to consider: commission fees, market impact or bid-ask spread and short-sell costs.

Commission costs

A commission cost is a service charge assessed by a broker or investment advisor for handling purchases and sales of securities in the market. The commission costs can include brokerage fees, exchange fees and other costs regarding the execution of trades. Do and Faff [56] showed that commission costs decreased significantly (on average an institutional investor paid 70 bps[6] in 1963, while in 2009 paid only 9 bps). Do and Faff [56] conducted their study between 2009 and 2012, and there is evidence that commission costs have declined further to 3 bps [60].

Bid-ask spread

The bid-ask spread is essentially the difference between the highest price that a buyer is willing to pay for a security, and the lowest price seller is willing to accept. Therefore, a share price will move between the bid and ask price (bid-ask bounce). This bid-ask bounce as a trading cost represents the possibility of a transaction occurring at a costlier price than the closing price, since it may not have been

[5]The bid-ask bounce is a specific situation when the price of a security bounces back and forth within a minimal range between the bid and ask price. This happens when trades are on both the bid and ask price, but no real movement in price.

[6]Basis points (bps) refers to a standard unit of measure in finance, 1 bps = 0.01%.

possible to buy or sell the security at the given close price. Do and Faff [56] refers to the bid-ask spread cost as the cost of the market impact.

Considering Gatev et al. [4]'s approach, we argue that the one-day delay does not fully represent the transaction costs. The lower returns reported on their work when waiting one day may also be due to missed profit opportunities. To ensure that no advantage is gained by choosing not to wait one day, we consider the bid-ask spread when opening and closing trade positions.

Short-sell Costs

Short selling a security means that an investor sells a security without owning the security. The short-seller borrows the security from a lender and sells the security in the open market, planning to repurchase it later for less money. The first step of short selling is to find a willing lender of a security. The lenders are typically brokers that act as lending agents to owners of securities that wish to borrow out their securities. Associated with short selling is usually a fee applied per annum payable over each pair trade's life, and it is this fee that we take into account in the transaction costs.

These costs mentioned are generally not fixed rates but vary depending on the security involved, the broker and market chosen and other factors.[7] Nevertheless, and for convenience, we assume that all costs are constant along all pairs and periods in our study. Table 4.5 summarises the costs assumed in this study.

Table 4.5: Transaction costs considered.

Commission Costs	Bid-ask Spread	Short-sell Costs
8 bps	20 bps	1% per annum

It should be noted that the values considered are conservative, taking into account the estimates of Do and Faff [56] and the descriptions presented above. From what is expected to yield reliable results, that tend to be more favourable in a real trading environment.

4.4 Evaluation Metrics

We check our experimental results based on three evaluation metrics: return on investment (ROI), Sharpe ratio (SR) and maximum drawdown (MDD).

4.4.1 Return on Investment

Return on investment (ROI) is an approximate measure of an investment's profitability. By definition, it is a ratio that compares the gain or loss from an investment relative to its costs, as follows:

$$\text{ROI} = \frac{\text{Net Return on Investment}}{\text{Cost of Investment}} \times 100\%. \tag{4.1}$$

[7] The most striking example of rate variation is perhaps the bid-ask spread, where the difference in the liquidity of securities and markets is the predominant factor.

Recalling the portfolio construction, where we always require an initial investment of $\$1$, it was thought to become simple at any point in time to directly interpret the return on our portfolio as the net profit of the investment. This feature alone is also fundamental to the reward sign design for our RL agent.

Something to remark and mentioned by Gatev et al. [4] is that the calculation of the return on the whole portfolio is a "nontrivial issue". A necessary precaution is to ensure that we only measure returns from the pairs trading strategy, not eventual gains from the increase in portfolio securities' value. To this end, please note that the portfolio consists of cash only and not of securities being traded, all securities bought or sold are only held while a position is open. This also ensures that the agent's behaviour is not influenced by market value changes, since it only measures the portfolio's value.

When calculating the returns on a portfolio of pairs, Gatev et al. [4] referred to two approaches: the return on committed capital (ROCC) and the fully invested return. Both can be found in the literature, with the former scaling portfolio payoffs by all pairs selected for trading, the latter considering only with the pairs that opened positions during the trading period. The former is more conservative and comes much closer to the reality of most investors. It considers the opportunity cost of hedge funds of having to commit capital to a strategy even if the strategy does not trade. For these reasons, this work considered the return on committed capital.

There is an additional consideration that is worth mentioning. In a realistic scenario, when the borrower returns the lender's security, the lender returns the cash collateral plus an interest.[8] Therefore, this interest should be integrated into the sum of net returns. However, it was neglected for the ease of calculation and represented a tiny portion of the net returns.

All the returns presented in this study are in a leveraged position, as explained in section 4.3.1 when referring to the fact that it is a self-funding strategy. On an unleveraged position, the initial capital would be the initial gross exposure, i.e., the long position's value plus the short position's value so that the returns would be slightly reduced.

4.4.2 Sharpe Ratio

The ROI is a handy performance evaluation metric, but there are many flaws with using this measure in isolation. The calculation of returns for specific strategies is not entirely straightforward, especially for not directional strategies such as market-neutral variants or strategies that use leverage. Also, if we are presented with two strategies possessing identical returns, how do we know which one contains more risk? This question and other strategy comparison and risk assessment issues motivated the use of the Sharpe ratio.

William F. Sharpe is a Nobel-prize winning economist, who developed the Sharpe ratio in 1966 (later updated in 1994) [61]. The Sharpe ratio became one of the most commonly cited statistics in financial analysis and the risk-adjusted performance metric of choice amongst investors. It aims at measuring the excess return of the risk-free rate per unit of volatility. Volatility is a measure of the price fluctuations of a portfolio (or an asset), typically referred to as risk. The annual Sharpe ratio is computed as

[8] Remember that in a short-selling, the borrower needs to pay the cash collateral to the lender.

$$SR_{\text{year}} = \frac{R^{\text{port}} - R_f}{\sigma_{\text{port}}} \times \text{annualisation factor.} \tag{4.2}$$

We proceed to describe each of the elements composing (4.2). R^{port} represents the expected daily portfolio returns. This may be calculated as the mean value of the portfolio returns, given by

$$R_t^{\text{port}} = \sum_{i=1}^{N} \omega_i R_t^i, \tag{4.3}$$

where ω_i and R_t^i represent the weight of the i-th pair in a portfolio of size N[9] and the corresponding pair's daily returns, respectively. The risk-free rate, R_f, represents the theoretical return rate of an investment with zero risks. This study follows the most common practice of using the interest paid on a three-month U.S. Treasury bill as the risk-free rate.[10] Table 4.6 illustrates the risk-free annualised rates considered for each tested period. The values were obtained by taking the daily average of the three-month treasury bill rate during the corresponding period. The data is collected from [62].

Table 4.6: Risk-free rates considered per test period.

2015	2016	2017	2018	2019
0.052%	0.316%	0.934%	1.936%	2.062%

The values in table 4.6 should be converted into daily returns to be consistent with the terms of (4.2).[11]

The term σ_{port} represents the portfolio's volatility, and it is a measure of risk, or uncertainty, of returns. It is defined as the square root of the portfolio variance, which depends not only on the standard deviations of each security in the portfolio but also on the correlations.[12] The calculation is done as follows

$$\sigma_{\text{port}} = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N} \omega_i \omega_j \text{Cov}_{i,\,j}} = \sum_{i=1}^{N}\sum_{j=1}^{N} \omega_i \omega_j \rho_{i,\,j} \sigma_i \sigma_j. \tag{4.4}$$

Last but not least, we need to define the annualisation factor. The way of defining it can have a massive impact on our estimations of the Sharpe ratio. The annualisation factor allows us to express the estimated daily Sharpe ratio in annual terms. Expressing the annualised Sharpe ratio is not only a prevalent practice, but it makes the estimate consistent with the duration of the test periods. The widespread technique of calculating the annual Sharpe ratio from the daily Sharpe ratio is by multiplying it by $\sqrt{252}$ since 252 is the number of trading days in a year. However, as pointed out by Lo [63], this is only possible under exceptional circumstances. Depending on the serial correlation of the portfolio's returns, the Sharpe ratios can be considerably smaller (in the case of positive serial correlation) or

[9] For an equal-weighted portfolio, $\omega_i = \frac{1}{N}$.

[10] The risk-free rate is hypothetical, as every investment has some risk associated with it. However, treasury bills are the closest investment possible for being risk-free. Although the U.S. government can default on debt obligations, the probability of this happening is very low.

[11] The risk-free rate for one year, $\left(1 + R_f^{\text{year}}\right) = \left(1 + R_f^{\text{daily}}\right)^{252}$, can be converted to daily returns by $R_f^{\text{daily}} = \left(1 + R_f^{\text{year}}\right)^{\frac{1}{252}} - 1$.

[12] A lower correlation between securities in a portfolio results in a lower portfolio variance.

larger (in the case of negative serial correlation). The only case in which the approximation is truly valid is if the portfolio returns are independently and identically distributed (IID). This assumption is overly restrictive and often violated by financial data. Therefore, for a more rigorous approximation, in this work, we propose to adopt the correction factor proposed in [63]. In this case, the portfolio returns serial correlation is measured, and a scale factor is applied in accordance. The scale factors are characterised in appendix D.

There are some limitations and disadvantages to the use of the Sharpe ratio, one of them being the assumption that the returns being used are normally distributed (i.e., Gaussian). Unfortunately, this is not the case in financial markets. The returns are the skewed opposite of the average because there are generally several unexpected spikes and drops in prices. Hence, the Sharpe ratio is poor at characterising tail risk[13]. Additionally, the volatility measurement assumes that price movements in either direction are equally risky, regardless of if its upside (big positive returns) or downside volatility. The most common alternative is the Sortino ratio, a variation of the Sharpe ratio that uses the standard deviation of negative portfolio returns. There are other alternatives, such as the Omega ratio proposed by Keating and Shadwick [64]. So, why using the Sharpe ratio? The first and probably most important reason is that it is widely used in finance and pairs trading literature. It is a metric of choice in similar research works [2, 4, 11, 12]. Secondly, the Sharpe ratio is independent of leverage. Since the risk-free rate is a minor factor, leveraging means multiplying both the numerator and numerator by a constant, which can be cancelled remaining the result constant. This property is fundamental, given that our portfolio is built on leverage.

4.4.3 Maximum Drawdown

The maximum drawdown (MDD) represents the maximum observed cumulative loss from a peak to a trough of the portfolio's value, before a new peak is attained. It is considered to be an indicator of downside risk for a given time frame. Formally, during a given investment period, where $P(t)$ is the total account balance of the portfolio, and T is the terminal time value, the MDD is computed as

$$MDD(T) = \max_{\tau \in (0,T)} \left[\max_{t \in (0,\tau)} \frac{P(t) - P(\tau)}{P(t)} \times 100\% \right]. \tag{4.5}$$

4.5 Conclusion

We started in this chapter by describing the dataset used and how we prepared and divided the data. We then give some insights into how each phase of the study was developed. We proceeded with a detailed description of how we approximate our simulation to a real trading situation and, finally, the chosen evaluation metrics.

[13]Tail risk is a form of portfolio risk that the possibility of an investment moving more than three standard deviations from the mean is larger than is shown by the normal distribution.

Chapter 5

Results

Up to this point, we had the opportunity to explore the worlds of pairs trading and deep reinforcement learning. We suggested alternatives to the traditional form of pairs trading that aim to strengthen this strategy. In the previous chapter, we outlined how implementation and simulations should be conducted. This chapter compiles the results obtained. Initially, we illustrate the outcome of the data preprocessing. Then the results collected in study phase 1. Furthermore, the results obtained in study phase 2 (DRL model) are exhibited

5.1 Data Cleaning

As a reminder for the reader, our dataset is composed of a set of 208 different ETFs. Logically, not all ETFs were available or traded during all periods considered (many ETFs are relatively recent and were created during the interval studied). Some of them also violate the required conditions imposed in section 4.1.3.

To illustrate the outcome of processing the original dataset, table 5.1 shows each step's effects in the process for each period considered. The first row shows the number of ETFs discarded due to missing values. This may be because the ETF is not traded for the full period or because there are sporadic missing values in the price series. The second row displays the number of ETFs that did not meet the liquidity condition imposed. The third row contains the ETFs with outliers. Here, on the left side, we have the number of ETFs that contained only one outlier and that their value has been corrected. On the right side, we have the number of ETFs with more than one outlier in their price series, and these were simply discarded. It is essential to clarify that the rows are presented in order of application, so for example, we only assess the liquidity requirement for the ETFs that succeeded in the first step (no issues in data). Finally, the last row shows the total percentage of ETFs' initial universe considered for that period.

Table 5.1: Data cleaning results.

Study Phase	**Study Phase 1**			**Study Phase 2**	
Period	**2013 - 2015**	**2014 - 2016**	**2015 - 2017**	**2011 - 2018**	**2012 - 2019**
# of ETFs with insufficient data	88	77	70	108	99
# of illiquid ETFs	18	17	18	14	21
# of ETFs with outliers *(repaired - removed)*	1 - 1	1 - 0	0 - 0	3 - 3	3 - 3
Total percentage of ETFs considered	**49%**	**55%**	**58%**	**40%**	**41%**

5.2 Study Phase 1

In this section, we focus on the analysis of the results related to study phase 1. If the reader wishes to revive how the study phase 1 is structured, we recommend looking again at figure 4.4. The periods under analysis correspond to those assigned to study phase 1 in table 4.2.

5.2.1 Pairs Selection

In this section, we will detail the pairs selection procedure. First of all, it is important to recall some key points. Our search space is not limited; we do not impose any restriction on possible pairs combinations (section 3.3.2). The selection process is composed of four criteria (section 3.3.3), and we want to assess the impact of each of them. Also, we want to understand the influence of the method chosen to calculate the spread. To this end, we propose to illustrate the number of filtered pairs for each condition for each of the proposed linear regressions.

Table 5.2 shows the number of pairs removed in each step. In the first row, we have the pairs that did not show themselves to be cointegrated. In the second row, the pairs that passed the cointegration test but missed the spread Hurst exponent criterion. In the third and fourth row, the portion of the pairs that passed the previous condition but failed the respective criterion (half-life and mean-crosses). Again, the results are presented sequentially. This means that each row represents the pairs eliminated from the subset resulting from the previous selection condition. Naturally, most pairs will be eliminated initially. This happens not only because cointegration is the first step, but also because it is the most restrictive.

Table 5.2: Pairs selection results.

Formation Period		**2013 - 2014**		**2014 - 2015**		**2015 - 2016**	
Linear Regression		**OLS**	**TLS**	**OLS**	**TLS**	**OLS**	**TLS**
Total Pair Combinations		5050		6441		7140	
Eliminated Pairs per Stage	1. Cointegration	5030	5031	6340	6346	7103	7106
	2. Hurst Exponent	0	0	1	1	0	0
	3. Half-life	5	5	8	8	4	4
	4. Mean-crosses	0	0	1	1	0	0
Selected Pairs		**15**	**14**	**91**	**85**	**33**	**30**

Looking at table 5.2, the first thing that becomes clear is the confirmation that the cointegration test has a profound impact since that is where most of the filtering takes place. It is also implied that some pairs are not elected, because their convergence period is not compatible with the trading period, therefore not verifying the half-life condition.

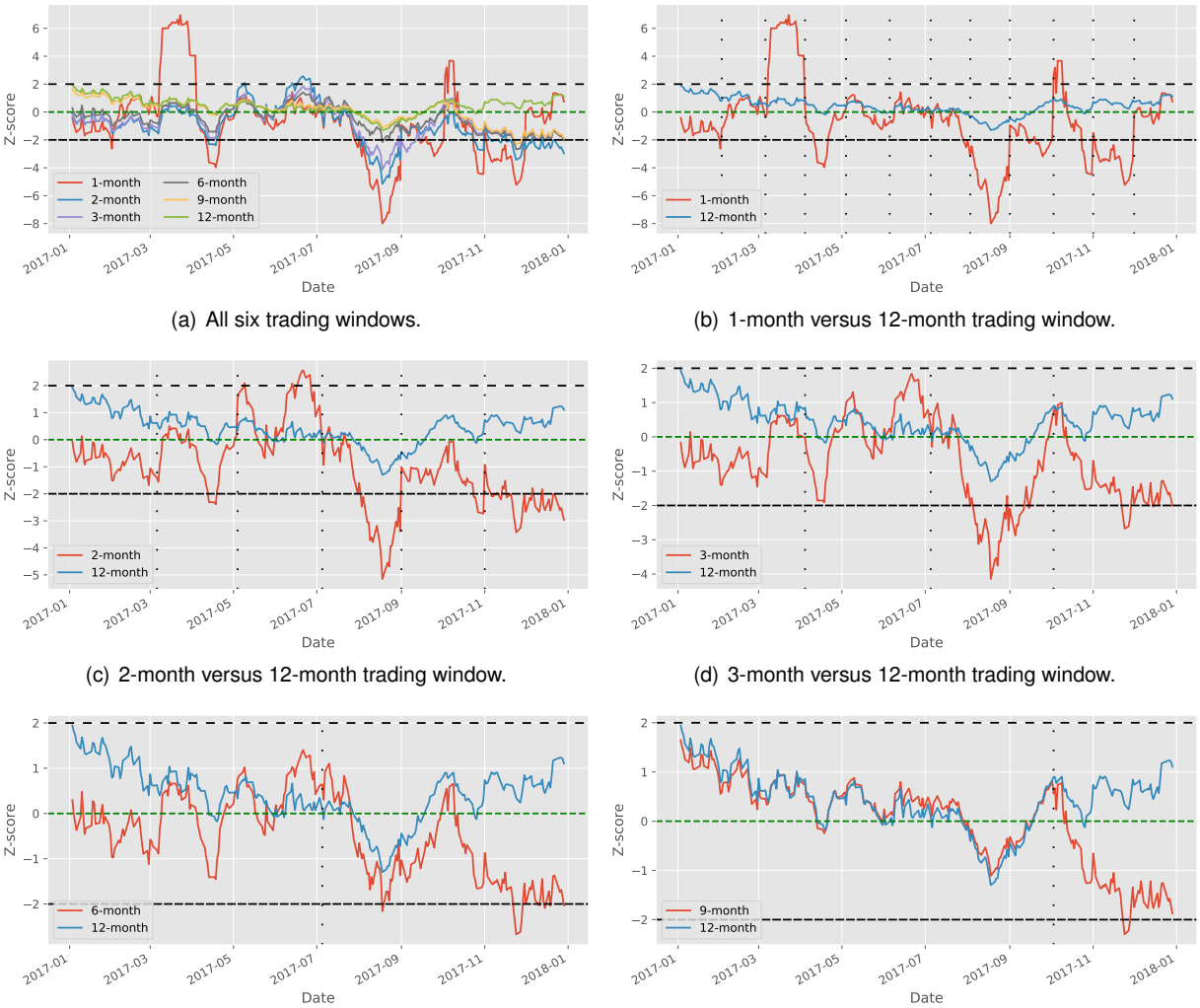
Regarding the spread Hurst exponent, the purpose of introducing it was to mitigate the multiple comparison problem and to identify the mean-reverting character of the pairs. However, this criterion had no influence (with one exception). Two interpretations may explain this lower relevance. On the one hand, the selected pairs guaranteed these conditions, or on the other hand, the Hurst exponent is not the best method to guarantee these conditions and alternative methods are needed. Finally, it should be noted that all pairs, with one exception, met the mean-crossing criterion. That said, in our scenario, both the Hurst exponent and the mean-crossing criterion turn out to be redundant.

Concerning the two linear regressions proposed, we note that the differences are minimal. This behaviour is not unfamiliar, as we had already seen that in practical terms, the hedge ratios obtained by OLS and TLS do not differ much. However, it can be seen that TLS in both three periods tends to select fewer pairs than OLS. In section 5.2.3, we will assess what real impacts this slight trend may translate.

5.2.2 Comparison of Window Sizes

Before we go into the trading results, we must remember that we want to study the influence of the window size. We experimented six cases as described in section 4.2.1, and in this section, we want to give the reader a taste of how different window sizes change the spread. Figure 5.1 illustrates exactly this, exemplifying the signal for each of the chosen sizes. The pair to use as an example is the DBO and PXE during the trading period from January 2017 to December 2017. This pair and this period were chosen arbitrarily, and similar behaviour can be found for all pairs and periods under study. The first subfigure shows the development of the Z-score values for all six windows simultaneously, with the remaining figures having each of the windows (red line) together with the 12-month reference trading window (blue line). It is crucial to have a reference to guarantee a visual comparison standard. The choice of the 12-month window specifically is because it is the largest window, corresponding to the total

trading period's coverage and the most "stable" signal.



(a) All six trading windows.

(b) 1-month versus 12-month trading window.

(c) 2-month versus 12-month trading window.

(d) 3-month versus 12-month trading window.

(e) 6-month versus 12-month trading window.

(f) 9-month versus 12-month trading window.

Figure 5.1: Comparison of applying the six different trading window sizes to the pair of ETFs, DBO and PXE, during the trading period of 2017.

Making a brief description of the figure, we have the usual horizontal lines representing each of the thresholds in the traditional model (threshold-based model) and which are essential as guidelines and for better interpretation of the signals' behaviour. The vertical lines loosely dotted indicate a trading window's end and the beginning of a new one. These lines always refer to the changing signal, since the 12-month reference signal is fixed and occupies the whole period. If the reader notices the figure 5.1(c), referring to the window size of 2-month, we have five vertical lines, representing six trading windows. This conduct should be familiar to the reader, as it is the one described in figure 4.3.

Figure 5.1(a) shows that the signals are quite different, almost suggesting that they belong to different pairs. However, with a closer look, we notice that the signal variations are similar, in the sense that when one is going up, the other is also going up. In essence, the order of magnitude of these variations is where they differ from signal to signal.

Looking now with a trading perspective, the signal with a 12-month trading window does not present any possibility of opening a position according to the traditional model, a situation that no longer happens

when the chosen trading window is shorter. The shorter the trading window, the greater the tendency for the number of possible positions to increase, as the spread signal becomes more sensitive to price series variations of the two ETFs considered for this pair.

These signals were extracted using the TLS method. As we have already pointed out more than once throughout this work, the differences in the signals extracted using OLS or TLS are much less visible.[1] However, these differences will have a significant impact as we will observe in the next section.

5.2.3 Trading Performance

In this section, we will focus on trading performance. We will analyse each of the windows sizes and each proposed linear regression and investigate the procedures that generated the most promising portfolios.

Table 5.3 unveils the results concerning the unseen data, the trading periods. This table describes each trading period's portfolio results when adopting any of the six window sizes and adopting both linear regressions. The rightmost column aggregates the information more concisely, including the average across all portfolios and periods for each linear regression. It should also be noted that the three evaluations metrics described in section 4.4 are highlighted, in order to differentiate them from the other statistics that we also consider relevant to present but are less critical.

[1] We find it redundant to present comparative charts between the two methods since visually the behaviour is similar.

Table 5.3: Results of the trading period for study phase 1.

Test Period		2015		2016		2017		AVG.	
Linear Regression		OLS	TLS	OLS	TLS	OLS	TLS	OLS	TLS
Trading Window Size	# of pairs	15	14	91	85	33	30	46	43
	1-month								
	ROI	**-4.53%**	**-4.09%**	**-3.75%**	**-1.85%**	**1.42%**	**2.22%**	**-2.29%**	**-1.24%**
	SR	**-0.95**	**-0.71**	**-0.62**	**-0.34**	**0.20**	**0.45**	**-0.46**	**-0.20**
	MDD	**-4.98%**	**-6.51%**	**-7.02%**	**-5.62%**	**-2.11%**	**-2.25%**	**-4.70%**	**-4.79%**
	% of profitable pairs	47%	50%	36%	42%	58%	53%	47%	48%
	# of total trades	161	148	990	958	377	329	509	478
	% of profitable trades	49%	49%	47%	49%	53%	54%	50%	51%
	2-month								
	ROI	**4.72%**	**8.39%**	**-7.64%**	**-7.10%**	**1.37%**	**4.12%**	**-0.52%**	**1.80%**
	SR	**1.08**	**1.64**	**-1.35**	**-1.24**	**0.19**	**1.05**	**-0.03**	**0.48**
	MDD	**-3.27%**	**-2.85%**	**-11.75%**	**-11.51%**	**-1.87%**	**-2.19%**	**-5.63%**	**-5.52%**
	% of profitable pairs	73%	71%	31%	32%	48%	60%	51%	54%
	# of total trades	90	90	588	568	203	197	294	285
	% of profitable trades	61%	63%	50%	51%	58%	63%	56%	59%
	3-month								
	ROI	**1.58%**	**2.83%**	**-8.67%**	**-8.06%**	**2.33%**	**2.78%**	**-1.59%**	**-0.82%**
	SR	**0.37**	**0.49**	**-1.20**	**-1.11**	**0.51**	**0.64**	**-0.11**	**0.01**
	MDD	**-2.94%**	**-3.71%**	**-11.72%**	**-11.28%**	**-1.98%**	**-1.45%**	**-5.55%**	**-5.48%**
	% of profitable pairs	40%	43%	31%	32%	58%	57%	43%	44%
	# of total trades	57	62	423	404	141	130	207	199
	% of profitable trades	54%	52%	44%	45%	55%	58%	51%	52%
	6-month								
	ROI	**-2.92%**	**-0.33%**	**-6.61%**	**-5.69%**	**2.29%**	**5.08%**	**-2.41%**	**-0.31%**
	SR	**-0.39**	**-0.02**	**-0.26**	**-0.24**	**0.49**	**1.33**	**-0.05**	**0.36**
	MDD	**-8.33%**	**-7.03%**	**-12.48%**	**-11.81%**	**-1.93%**	**-1.64%**	**-7.58%**	**-6.83%**
	% of profitable pairs	47%	57%	43%	41%	63%	70%	51%	56%
	# of total trades	45	39	277	252	86	83	136	125
	% of profitable trades	56%	54%	60%	58%	57%	64%	58%	59%
	9-month								
	ROI	**-2.27%**	**-0.68%**	**-5.65%**	**-4.99%**	**1.47%**	**2.50%**	**-2.15%**	**-1.06%**
	SR	**-0.41**	**-0.09**	**-0.05**	**-0.02**	**0.19**	**0.48**	**-0.09**	**0.12**
	MDD	**-7.16%**	**-7.24%**	**-13.59%**	**-12.71%**	**-2.74%**	**-2.89%**	**-7.83%**	**-7.61%**
	% of profitable pairs	40%	36%	47%	46%	45%	47%	44%	43%
	# of total trades	39	36	207	191	80	73	109	100
	% of profitable trades	51%	50%	57%	54%	60%	62%	56%	55%
	12-month								
	ROI	**0.92%**	**-1.94%**	**-0.05%**	**0.67%**	**1.02%**	**2.44%**	**0.63%**	**0.39%**
	SR	**0.19**	**-0.26**	**0.00**	**0.01**	**0.05**	**0.47**	**0.08**	**0.07**
	MDD	**-4.59%**	**-6.65%**	**-12.70%**	**-12.40%**	**-2.48%**	**-2.51%**	**-6.59%**	**-7.19%**
	% of profitable pairs	53%	36%	47%	48%	39%	43%	46%	42%
	# of total trades	29	27	152	144	54	50	78	74
	% of profitable trades	62%	59%	59%	58%	59%	66%	60%	61%

The first impression we get is that overall the results are poor. One of the main reasons behind this low profitability of pairs trading is that our results consider realistic transaction costs described in section 4.3.3. This statement comes in line with the findings of Do and Faff [56], which states that after controlling for costs and accounting for systematic risks, the potential profits documented in the literature are not attainable. Nevertheless, the authors still mention that pairs trading remains profitable

in a relatively small number of refined versions but at much-diminished levels, as shown in the table above. On the other hand, Do and Faff [7] and Rad et al. [24] state that pairs trading performs strongly during periods of prolonged turbulence (for example the global financial crisis of 2007–2008), a situation that does not happen in the periods considered, with the market behaving positively.[2] However, we can still draw many conclusions from our results and evaluate the two main factors under study (window size and linear regression).

Focusing now only on the results for the two linear regressions, we observe that for all portfolio combinations of the six discrete window sizes and year of study, the trading signals made with TLS method are better than those made with OLS method. There is only one exception for the 12-month window and 2015, which then largely influences the average. In this case, when we declare that the results are better, we mean whether the investor is only interested in the highest possible ROI, or is also interested in the incurred risk, SR. If we include the MDD and the other statistics, this ascendancy is no longer so overwhelming but remains in most cases. The reason for this superiority of TLS is based on the difference between the hedge ratios of the two methods, as shown in section 3.2. These results confirm all the evidence, proving that TLS best captures the ETFs' functional relationship establishing the pair.

Concluding the ideal trading window size is a much less trivial matter, as there is no clear winner. In absolute terms and gazing at the average over the three years, the 2-month window for TLS shows the best results in both ROI and SR with 1.80% and 0.48 respectively. However, this result is closely linked to an exceptional performance in 2015, having obtained the best result for all portfolio combinations. When we include the OLS method, this window's performance is no longer as satisfactory. If we look at it from a consistency perspective, i.e. for both linear regressions and all years, the 12-month window can be considered the superior one, as it achieves positive profits on average for both linear regressions. It is clear the importance and influence of the trading window length on trading performance, not being a minor factor in optimising the pairs trading strategy parameters.

In an additional note and confirming the point raised in the previous section, the total number of trades in the shorter windows is much higher, decreasing naturally with the window's length increment. This behaviour is persistent throughout all the years as well as when we apply the average.

Pairs performance is very much associated with the percentage of profitable trades, so it was interesting to reflect on how we could increase this percentage. One suggestion would be not to allow opening positions too close to the trading window's end. This is more significant in shorter trading windows as it can happen more often. Intuitively, by opening a position close to the date set for the window closure, in addition to the associated transaction costs, we are probably not giving time for the spread to converge, and losses are incurred. One possible way to mitigate this problem would be, for example, to limit the opening of positions to n days from the ending of the window (variable n can be optimised).

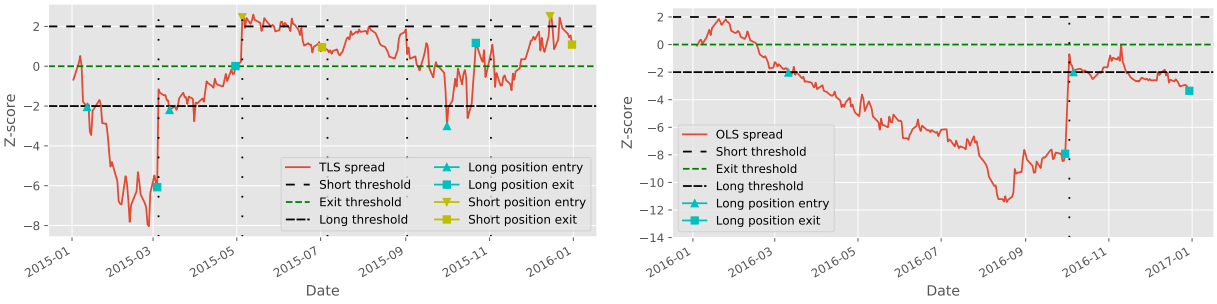
One last point to highlight is related to the percentage of profitable pairs in the portfolios, and that confirms that it is not a straightforward decision the trading window size to use. If we look solely to this metric, the 6-month window presents on average the highest percentage of profitable pairs in the

[2]The historical returns of the S&P 500 for the years under study were 1.31%, 21.94% and 11.93%, respectively. This index can be seen as a proxy for the state of the market and economy and is, therefore, a useful reference.

portfolio with 51% and 56% for OLS and TLS, respectively. However, this window cannot be profitable. We can confirm that it is the presence of pairs with an abysmal performance that explains these results. By contrast, we have the example of the 3-month window in 2015, wherewith only 40% (OLS) and 43% (TLS) of pairs being profitable, ROI and SR are both positive.

5.2.4 Extreme Cases

It is interesting to complement the results with the indication of the extreme cases, i.e. the best and the worst pair of the studied portfolios. The best pair was found in 2015 consisting of ETFs, XOP and IEZ, with a spread made by the TLS method and the 2-month trading window. Under these conditions, the pair got an ROI of 58.70% and an SR of 3.19. Once again, we were able to observe the impact that the window size has on the results, and for this same pair, for TLS and this test period, by varying the window we were able to go from an ROI of -1.38% (3-month) to 58.70% (2-month). In turn, the worst pair was found in 2016 consisting of ETFs, SLVP and AGQ, with a spread made by OLS method and for the 9-month trading window. The pair had an ROI of -69.03% and an SR of -0.11. The same behaviour happens again among the window sizes, where the 1-month window got an ROI of 5.29%.[3] Figure 5.2 illustrates the examples described.



(a) Extreme positive case.

(b) Extreme negative case.

Figure 5.2: Representation of the extreme cases in the portfolios previously studied - the best pair (left) and the worst pair (right).

We proceed to inspect each of the figures. In the extreme positive case, we observe that five positions were established, four of which were profitable. The position that did not make a profit was the first, demonstrating the importance of defining the windows since the spread was diverging and the recalibration of the parameters allowed the "recovery" and become very profitable. Regarding the extreme negative case, the divergence of the spread justifies the results. This example came from a pair with the spread made by the OLS method. For the same period (2014 - 2015 formation period and 2016 trading period), the TLS method does not select this pair, proving once again this method's greater robustness.

5.2.5 Daily Z-score Distribution

To take our analysis one step further, we decided to look at the Z-score values' daily distribution to understand if it is possible to detect some pattern to interpret the results. We think it is relevant to add

[3] If the reader is interested in knowing which segments or categories each ETF belong to, we invite to inspect appendix C.

this analysis, as we cannot find anything similar in the literature, and it is another contribution of our study to the field.

Table 5.4 presents the two factors that we think could contribute to our analysis, the Z-score mean and the percentage of observations more than two standard deviations away from the mean.[4] This table is organised in a similar way to table 5.3, in the sense that it presents the results for the various portfolio combinations previously mentioned. In the rightmost column, we once again exhibit the average across all combinations in that row. However, in the mean Z-score, this average is done in absolute terms, because here we are interested in understanding if our distribution deviates to either direction from the theoretical mean (zero).

Table 5.4: Results of the daily distribution of Z-score values.

Test Period		**2015**		**2016**		**2017**		**AVG.**	
Linear Regression		**OLS**	**TLS**	**OLS**	**TLS**	**OLS**	**TLS**	**OLS**	**TLS**
Trading Window Size	**1-month**								
	Mean Z-score	-0.060	0.417	0.340	0.137	-0.310	-0.498	0.237	0.351
	Observations outside $\pm$ 2	41%	39%	59%	58%	37%	35%	46%	44%
	2-month								
	Mean Z-score	-0.244	0.309	0.267	0.104	-0.137	-0.491	0.216	0.301
	Observations outside $\pm$ 2	37%	37%	41%	41%	34%	34%	37%	37%
	3-month								
	Mean Z-score	-0.254	0.439	0.567	0.237	-0.125	-0.641	0.315	0.439
	Observations outside $\pm$ 2	36%	39%	52%	52%	35%	34%	41%	42%
	6-month								
	Mean Z-score	0.185	0.411	0.780	0.269	-0.080	-0.599	0.348	0.426
	Observations outside $\pm$ 2	41%	43%	58%	58%	35%	33%	45%	45%
	9-month								
	Mean Z-score	-0.315	0.180	0.790	0.073	0.134	-0.562	0.413	0.272
	Observations outside $\pm$ 2	36%	38%	51%	52%	32%	31%	40%	40%
	12-month								
	Mean Z-score	-0.008	0.758	0.740	0.035	0.274	-0.581	0.341	0.458
	Observations outside $\pm$ 2	39%	41%	59%	60%	31%	30%	43%	44%

By analysing table 5.4, it is interesting to note that the percentage of observations more than two standard deviations away from the mean can be divided into two broad groups (not taking into account the average - rightmost column): (i) less than 43% and (ii) more than 52%.

We can find a correlation between a high percentage of observations outside the thresholds for open-

[4] The reader should remember that the thresholds for opening a position are set at $\pm 2\sigma$, where σ is the standard deviation.

ing positions, i.e., belonging to the group (ii), and poor trading performance. If we add a larger magnitude of the mean deviation of the distribution and the theoretical mean, this correlation becomes even more evident. This interdependence is especially true in 2016 when performance is worse, corroborated by a percentage value always belonging to the group (ii) (with one exception). We can confirm that this higher percentage is synonymous of more divergent pairs in the trading period.

The behaviour described is once again evident in 2017 when the percentage is always below 37%, proving the hypothesis, as it is the year with the best trading performance of the three study periods considered.

2015 is the year with the vastest diversity, and the points presented are again supported. The 1-month and 6-month windows have the worst performance concerning the other windows in that year, with the highest percentages of observations outside more than two standard deviations. If we look at the concrete example of the 12-month window, in terms of percentage the difference between the two linear regressions is not very considerable, but in terms of trading performance, the difference is immense. Here, we believe that the mean deviation of the distribution can explain the difference, and for OLS, the mean is almost precisely the theoretical mean of zero.

To deepen the study initiated here and adding this information to the system to perform some filtering based on, for example, an additional validation period, it could be quite promising.

Figure 5.3 includes a visual example of the daily Z-score distributions for the TLS method and the six different windows studied. Presenting the distributions for all the years would be redundant, so we chose the year of 2016 based on the fact that it is the most representative year in terms of numbers of pairs and it is also the year where we mostly focus our analysis.

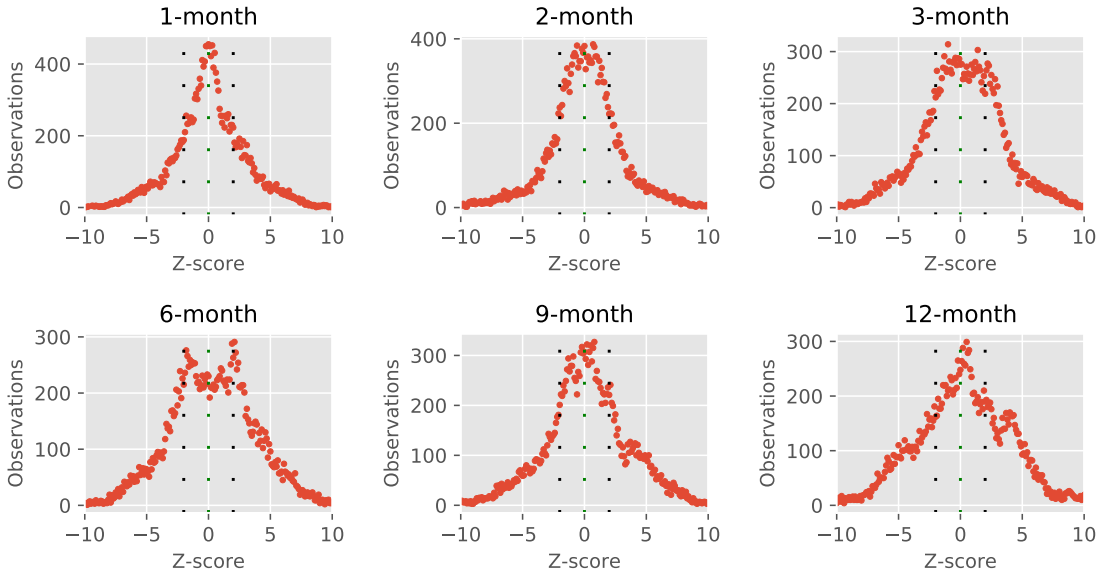


Figure 5.3: Daily distribution of Z-score values for the six different trading windows sizes for the portfolios consisted of trading signals obtained by the TLS method in 2016.

5.3 Study Phase 2

In this section, we will look at the results of study phase 2. The reader should remember that in the design of this study phase, the number of possibilities for experimentation was limited due to the painful training process of our DRL model. As described in section 4.2.2, our options are to adopt the best linear regression, limit the number of pairs to the top 10 and estimate the window size most reliable. This information is collected based on the performance throughout the study phase 1. In the linear regression examination, we discussed that TLS stood out as unequivocally better than OLS in the previous section. Concerning the size of the trading windows, this decision is not clear. In absolute terms, the 2-month window appears to be the best. However, if we look carefully at the results of study phase 1, and in line with our analysis, the 12-month window is more reliable, so our choice falls on this length. In short and following figure 4.5, the trading signals are made through the TLS method, the top 10 ETFs pairs are chosen, a 12-month trading window is applied and the two trading models proposed are implemented for comparison (our DRL model and the standard threshold-based model). The periods under analysis correspond to those assigned to study phase 2 in table 4.2.

5.3.1 Training

Before we get into the trading performance and testing, it is crucial to confirm that our DRL algorithm is trained well. In section 3.4.4, we pointed out that the only way to determine how training is going on in DRL is through the average rewards collected for each episode. So in this section, we will present those same results. This training is done during the two formation periods considered, i.e. from Jan 2011 to Dec 2017 and Jan 2012 to Dec 2018 (figure 4.2). Nevertheless, before we go to the training demonstrations, there are two more points that we want to touch.

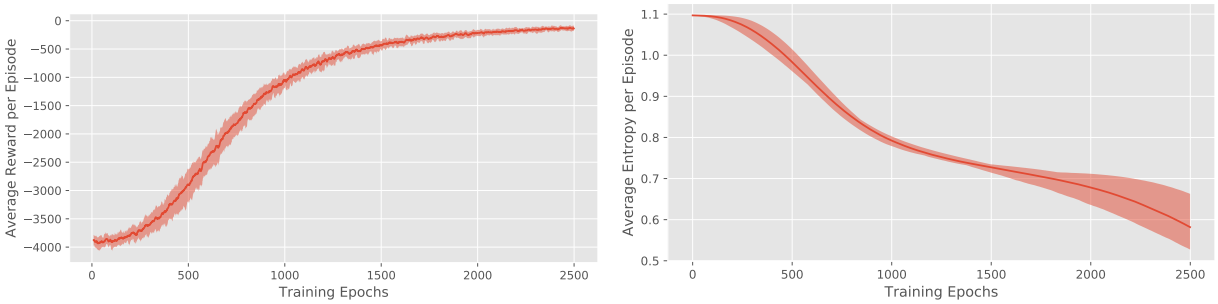
In addition to rewards, plotting the entropy[5] provides valuable information about how well the model is performing. First of all, the model has three outputs (one for each action). If each action is equally probable, then entropy is $-3 \cdot \frac{1}{3} \cdot \ln \frac{1}{3}$[6], roughly 1.1. As long as the logged entropy keeps this value, the model does not perform better than randomly picking an action. It is good to calculate a few cases to give the reader a feeling for the probabilities and the related entropy. For example, when one action's probability is three times more probable than the rest, the entropy is $-(2 \cdot 0.2 \cdot \ln 0.2 + 0.6 \cdot \ln 0.6)$ which is 0.95. Doing the same reasoning for eight times more becomes 0.64, and so on.

Henderson et al. [52] reported that despite all the efforts already made in DRL and applied in this work, one of the biggest concerns is still the large variance in the results across trials and random seeds. This variance comes from the environment stochasticity or stochasticity in the learning process (e.g. random weight initialisation). To mitigate this issue, the authors suggest performing multiple trials with different random seeds to show the reproducibility of DRL. Therefore, we run five trials for each experiment, and all results shown from now on are an average of these trials.

Figure 5.4 and 5.5 illustrate on the left the development of the average sum of rewards per episode

[5] As a reminder, the entropy corresponds to the spread of action probabilities.
[6] Recall that the entropy used in the actor loss function is the Shannon entropy.

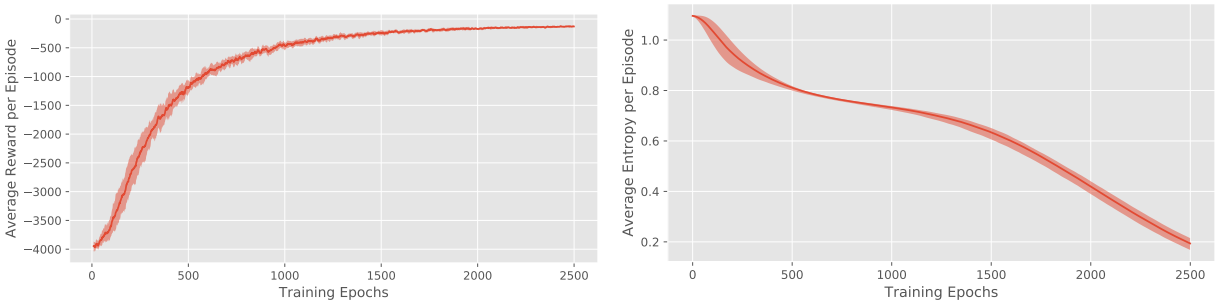
and the right the average entropy per episode. Presenting all ten pairs for each test period was inconceivable, so we decided to exhibit one arbitrary pair per period.



(a) Rewards learning curve.

(b) Entropy learning curve.

Figure 5.4: Learning curves for the pair of ETFs, UCO and DBE, in the training period from Jan 2011 to Dec 2017.



(a) Rewards learning curve.

(b) Entropy learning curve.

Figure 5.5: Learning curves for the pair of ETFs, UGL and AGQ, in the training period from Jan 2012 to Dec 2018.

The solid line represents the five trials' average and the filled region between the maximum and minimum values for rewards and entropy trials. A simple moving average over the previous ten training epochs was made to smooth the rewards curve. The original curve was quite noisy due mainly to the exploratory component. The algorithm keeps trying new actions, and some of them will fail.

The rewards are always negative because of the penalty we give to the system when a position is not open. However, it is essential to remember that the system does not "see" the rewards in these values, due to the standardisation made during the training, but here we are displaying the absolute rewards. We can find that the average sum of rewards per episode steadily increased, indicating that the DRL model is appropriately trained.

In both entropy plots, we can see that it starts just below 1.1, and then it starts to drop after a while. The drop in figure 5.5(b) is faster and sharper than in figure 5.4(b) because this pair arrives more quickly to the asymptotic value of rewards. We also see that the entropy value drops to lower values for figure 5.5(b), which means more certainty for the agent regarding the actions to take. We assume that for this pair, the agent learned the optimal behaviour faster.

5.3.2 Trading Performance

Having described the design structure and how training is conducted, we may analyse the test results, which resemble a practical trading environment. At this study phase, having a smaller number of pairs (10) allows us to analyse each pair's performance before looking at the portfolios as a whole. Thus, table 5.5 shows the average performance measures for each pair in the trading period of 2019. The results are set side by side with the results obtained using the standard threshold-based model for comparison purposes. A similar table can be found in appendix E for the trading period of 2018.

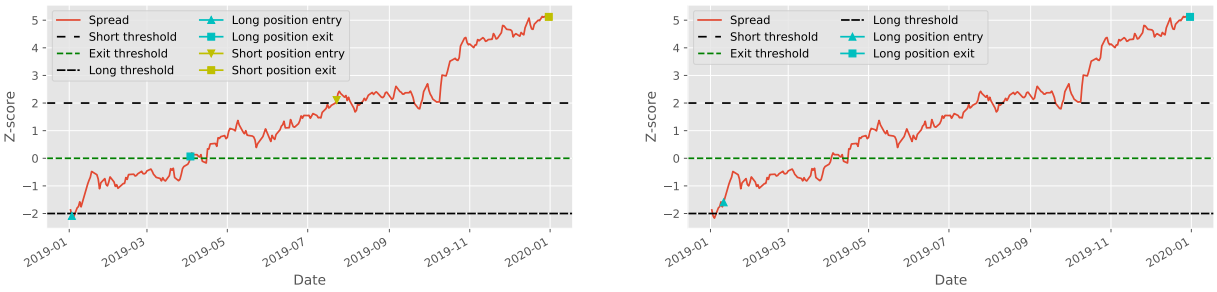
Table 5.5: Comparison of trading performance per pair for the trading period of 2019.

Trading Model		**Standard Trading Model**			**DRL Trading Model**		
Evaluation Metrics		**ROI**	**SR**	**MDD**	**ROI**	**SR**	**MDD**
Pairs	**DBP / AGQ**	-7.30%	-1.52	-7.77%	-12.94%	-1.96	-13.27%
	DGP / AGQ	-15.64%	-0.68	-26.55%	27.16%	1.01	-20.16%
	UGL / AGQ	-10.60%	-0.78	-15.58%	21.66%	1.32	-8.76%
	NLR / CGW	-2.31%	-0.56	-14.33%	32.74%	3.56	-3.62%
	UGA / DBE	-4.45%	-1.13	-8.99%	-1.08%	-0.26	-15.06%
	UGA / DBO	-5.58%	-0.95	-9.35%	9.12%	0.37	-16.39%
	SIVR / DBP	9.92%	0.65	-11.73%	2.48%	0.03	-11.46%
	SLV / DBP	10.02%	0.64	-11.93%	2.48%	0.03	-11.67%
	DBP / UGL	1.77%	-0.16	-1.07%	-1.98%	-1.48	-3.21%
	SIVR / DGL	9.50%	0.53	-13.26%	2.93%	0.06	-13.19%

We can see that the DRL model performs better than the standard trading model in only half of the pairs. However, solely three pairs could not be profitable contrary to the six pairs in the standard trading model. Of these three pairs that were not profitable, the losses were not very considerable in two of them. In terms of earnings, the proposed model also managed to have the pairs with the most significance. The pair with the highest profit using the proposed method was NLR and CGW (32.74%); it also shows one of the biggest differences between the DRL model and the standard trading model (-2.31%) - only surpassed by the pair DGP and AGQ. Let us look at this particular pair and compare both models' behaviour, as described in figure 5.6.

We found that both models' behaviour is similar at an early stage, with the opening of the long position in both models (in the DRL model, this opening happens slightly later). However, what happens next is a super simplistic behaviour on the part of the proposed model of not performing any action during the rest of the period. It can be said that this was the ideal behaviour to have for this pair. We suspect that since the agent's model states are defined as the difference between the Z-scores, the agent does not detect any reversal pattern of the signal's upward trend, hence deciding not to close the long position earlier.

Additionally, we can confirm that where the proposed model stands out the most is in divergent pairs, proven by the DGP/AGQ and UGL/AGQ pairs' performance in table 5.5. Again here it emerges that the



(a) Positions set by the standard trading model.

(b) Positions set by the DRL model.

Figure 5.6: Comparison of position setting behaviour for both models for the pair of ETFs, NLR and CGW, in the 2019 trading period.

agent does not "see" the signal in the same way as the traditional model does, not knowing if it is, in fact, diverging or has an expected behaviour (mean-reversion). This behaviour shows indications that a merger between the two systems may be appealing.

Turning now the analysis to the portfolios and adding the trading period of 2018, the average performance measures obtained are shown in table 5.6. The results are displayed for each trading period, with the other three metrics that had already been used in the previous study phase being added to the three evaluation metrics. In the rightmost column, the average of the two periods is presented as usual.

Table 5.6: Results of the trading period for study phase 2.

Trading Model	**Standard Trading Model**			**DRL Trading Model**		
Test Period	**2018**	**2019**	**AVG.**	**2018**	**2019**	**AVG.**
ROI	**6.28%**	**-1.47%**	**2.41%**	**1.54%**	**8.26%**	**4.9%**
SR	**0.81**	**-0.53**	**0.14**	**0.09**	**1.04**	**0.57**
MDD	**-6.00%**	**-5.43%**	**-5.72%**	**-7.64%**	**-4.79%**	**-6.22%**
% of profitable pairs	60%	40%	50%	40%	70%	35%
# of total trades	12	11	12	10	10	10
% of profitable trades	83%	45%	64%	40%	70%	35%

We can see that in 2018 the proposed model was profitable, but its performance was below the standard trading model, showing some indications of instability in the execution of the DRL model. Nevertheless, on average, our proposed model managed to be superior to the standard trading model for both periods, but it was not as disruptive as we believed it could be.

We can also see an interesting detail of always opening only one position per pair. Although we have no concrete reason for this pattern, we suspect that maybe the transaction costs may have some influence.

On a final note, we can see that the SR is not very expressive for two reasons. The first is that risk-free rates in these periods are high (table 4.6). The second is the higher volatility that the proposed model originated, as evidenced by the higher MDD on both tables. This higher volatility is a disadvantage of our model, as it is associated with a higher risk. If we add the SR in addition to the total profit as an objective function, we can build a more optimised trading pairs system.

Chapter 6

Conclusions and Future Work

This work is dividing into two main phases. In study phase 1, we focused on the impact of two key features (trading signal extraction and window size) on pairs trading profitability. In study phase 2, we investigate if it is possible to train a deep reinforcement learning model to optimise the strategy beyond the traditional way.

Starting by revisiting the first part, we collected pairs without any restriction in the search space (the total universe of ETFs considered) through a set of rules structured around the cointegration test. Next, we experimented how the results varied according to the spread and the methods used. Therefore, we defined different spreads using OLS and TLS methods and explored six different window sizes to find the most reliable one by varying the formation window and the trading window. The results showed that for all six window sizes and periods studied, TLS spread was better than the OLS spread. There was no clear winner in terms of the optimal window size, but we can highlight the 2-month trading window like the one with the best absolute results and the 12-month trading window as the most reliable. In any case, we opened a door for a factor to be more considered in the strategy's definition, given its impact on the results.

Turning to the second part, we designed and implemented an advantage actor-critic agent to make pairs trading decisions. Using the previous conclusions, we check whether our DRL model is trained well. We discovered that the average rewards steadily increased at each epoch and that the entropy dropped, confirming that the DRL model was learning. Based on these results, we discovered that our proposed model in the test periods outperforms the traditional pairs trading strategy on average for both out-of-sample datasets. Our model's great advantage was demonstrated in moments when the pair ends up diverging (the most significant risk associated with the pairs trading strategy) in unseen data managing to be still profitable. Although all efforts to reduce variance and make our model more stable, the noisy nature of financial data and the algorithm's tendency to get stuck in a local optimum made our task quite complicated. These reasons ultimately meant that the results were not as disruptive as we had anticipated they would be. However, we want to highlight one point. To date, we have not seen another application of a stochastic policy-based RL approach to pairs trading (all literature is based on Q-learning variants), so we hope to have contributed to this group of algorithms gaining space.

Train and deploy large scale end-to-end DRL trading algorithms is still in its infancy in quantitative trading, but we believe it is part of the future. DRL has dramatically improved traditional ML in the game space - AlphaGo is probably the most famous example - proving that the RL framework is quite versatile. This type of learning opens up possibilities to customise environments and train algorithms and reward functions for solving unique computational finance applications tasks effectively - Kolm and Ritter [65] presented possible new perspectives - that traditional supervised and unsupervised learning models cannot achieve.

6.1 Future Work

We can follow this work in multiple directions. Regarding pairs trading, although it has been a known strategy for several years, there is still room for further exploration. When it comes to DRL, a lot can still be done.

Improving the pairs trading framework:

- Use other statistical methods such as the Kalman filter and error-correction model to diversify the spreads used.
- Find more advanced time window optimisation mechanisms that can incorporate the dataset and model in use.

Improving the DRL model:

- Explore actor-critic approaches with continuous action space. For example, this could allow the agent to be free to determine what portion of the initial investment to allocate to each position. Besides, this could also allow exploring the opening of multiple positions simultaneously to, on the one hand, mitigate the risk/loss or, on the other hand, increase earnings.
- Profit was set as the objective function in this study. It could be interesting to add other performance indicators (such as risk-adjusted indicators).
- New algorithms are continually emerging and becoming increasingly reliable and stable. We want to highlight two that caught our attention: Trust Region Policy Optimization (TRPO) and Proximal Policy Optimization (PPO). It is necessary to assess their suitability for financial markets.

Bibliography

[1] R. C. Cavalcante, R. C. Brasileiro, V. L. F. Souza, J. P. Nobrega, and A. L. I. Oliveira. Computational Intelligence and Financial Markets: A Survey and Future Directions. *Expert Systems with Applications*, 55:194–211, Aug. 2016. ISSN 0957-4174. doi: 10.1016/j.eswa.2016.02.006.

[2] M. Avellaneda and J.-H. Lee. Statistical arbitrage in the US equities market. *Quantitative Finance*, 10(7):761–782, Aug. 2010. ISSN 1469-7688. doi: 10.1080/14697680903124632.

[3] S. Mallaby. Learning to love hedge funds. *The Wall Street Journal*, 2010.

[4] E. Gatev, W. N. Goetzmann, and K. G. Rouwenhorst. Pairs Trading: Performance of a Relative-Value Arbitrage Rule. *The Review of Financial Studies*, 19(3):797–827, Oct. 2006. ISSN 0893-9454. doi: 10.1093/rfs/hhj020.

[5] R. J. Elliott, J. V. D. Hoek *, and W. P. Malcolm. Pairs trading. *Quantitative Finance*, 5(3):271–276, June 2005. ISSN 1469-7688. doi: 10.1080/14697680500149370.

[6] S. C. Andrade and M. S. Seasholes. *Understanding the Profitability of Pairs Trading*. 2005.

[7] B. Do and R. Faff. Does Simple Pairs Trading Still Work? *Financial Analysts Journal*, 66(4):83–95, July 2010. ISSN 0015-198X, 1938-3312. doi: 10.2469/faj.v66.n4.1.

[8] S. Mudchanatongsuk, J. A. Primbs, and W. Wong. Optimal pairs trading: A stochastic control approach. In *2008 American Control Conference*, pages 1035–1039, Seattle, WA, June 2008. IEEE. ISBN 978-1-4244-2078-0. doi: 10.1109/ACC.2008.4586628.

[9] Z. Zeng and C.-G. Lee. Pairs trading: optimal thresholds and profitability. *Quantitative Finance*, 14 (11):1881–1893, Nov. 2014. ISSN 1469-7688, 1469-7696. doi: 10.1080/14697688.2014.917806.

[10] S. Fallahpour, H. Hakimian, K. Taheri, and E. Ramezanifar. Pairs trading strategy optimization using the reinforcement learning method: a cointegration approach. *Soft Computing*, 20(12):5051–5066, Dec. 2016. ISSN 1433-7479. doi: 10.1007/s00500-016-2298-4.

[11] T. Kim and H. Y. Kim. Optimizing the Pairs-Trading Strategy Using Deep Reinforcement Learning with Trading and Stop-Loss Boundaries, 2019.

[12] S. M. Sarmento and N. Horta. Enhancing a Pairs Trading strategy with the application of Machine Learning. *Expert Systems with Applications*, 158:113490, Nov. 2020. ISSN 09574174. doi: 10.1016/j.eswa.2020.113490.

[13] J. Goldkamp and M. Dehghanimohammadabadi. Evolutionary multi-objective optimization for multivariate pairs trading. *Expert Systems with Applications*, 135:113–128, Nov. 2019. ISSN 0957-4174. doi: 10.1016/j.eswa.2019.05.046.

[14] davidsilver.uk. "UCL Course on RL". [Online]. Available: `https://www.davidsilver.uk/teaching/`. [Accessed: 21 September 2020].

[15] R. S. Sutton and A. G. Barto. Reinforcement Learning: An Introduction. page 352, 1998.

[16] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing Atari with Deep Reinforcement Learning. *arXiv:1312.5602 [cs]*, Dec. 2013. arXiv: 1312.5602.

[17] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, Feb. 2015. ISSN 1476-4687. doi: 10.1038/nature14236. Number: 7540 Publisher: Nature Publishing Group.

[18] C. Krauss. Statistical Arbitrage Pairs Trading Strategies: Review and Outlook. *Journal of Economic Surveys*, 31(2):513–545, 2017. ISSN 1467-6419. doi: 10.1111/joes.12153.

[19] Z. Chen and F. Li. Empirical Investigation of an Equity Pairs Trading Strategy. *Management Science*, page 21, 2017.

[20] R. F. Engle and C. W. J. Granger. Co-Integration and Error Correction: Representation, Estimation, and Testing. *Econometrica*, 55(2):251–276, 1987. ISSN 0012-9682. doi: 10.2307/1913236. Publisher: [Wiley, Econometric Society].

[21] S. Johansen. Statistical analysis of cointegration vectors. *Journal of Economic Dynamics and Control*, 12(2-3):231–254, June 1988. ISSN 01651889. doi: 10.1016/0165-1889(88)90041-3.

[22] D. A. Dickey and W. A. Fuller. Distribution of the Estimators for Autoregressive Time Series With a Unit Root. *Journal of the American Statistical Association*, 74(366):427, June 1979. ISSN 01621459. doi: 10.2307/2286348.

[23] G. Vidyamurthy. *Pairs Trading: Quantitative Methods and Analysis*. John Wiley & Sons, Aug. 2004. ISBN 978-0-471-46067-1. Google-Books-ID: FTFuPFx0hdcC.

[24] H. Rad, R. K. Y. Low, and R. Faff. The profitability of pairs trading strategies: distance, cointegration and copula methods. *Quantitative Finance*, 16(10):1541–1558, Oct. 2016. ISSN 1469-7688. doi: 10.1080/14697688.2016.1164337.

[25] A. Mikkelsen. Pairs trading: the case of Norwegian seafood companies. *Applied Economics*, 50 (3):303–318, Jan. 2018. ISSN 0003-6846. doi: 10.1080/00036846.2017.1321837. Publisher: Routledge _eprint: https://doi.org/10.1080/00036846.2017.1321837.

[26] N. Huck and K. Afawubo. Pairs trading and selection methods: is cointegration superior? *Applied Economics*, 47(6):599–613, Feb. 2015. ISSN 0003-6846. doi: 10.1080/00036846.2014.975417.

[27] M. Clegg and C. Krauss. Pairs trading with partial cointegration. *Quantitative Finance*, 18(1):121–138, Jan. 2018. ISSN 1469-7688. doi: 10.1080/14697688.2017.1370122.

[28] R. Aguilar-Rivera, M. Valenzuela-Rendón, and J. J. Rodríguez-Ortiz. Genetic algorithms and Darwinian approaches in financial applications: A survey. *Expert Systems with Applications*, 42(21): 7684–7697, Nov. 2015. ISSN 0957-4174. doi: 10.1016/j.eswa.2015.06.001.

[29] K. Gupta and N. Chatterjee. Selecting stock pairs for pairs trading while incorporating lead–lag relationship. *Physica A: Statistical Mechanics and its Applications*, page 124103, Jan. 2020. ISSN 0378-4371. doi: 10.1016/j.physa.2019.124103.

[30] E. Chan. *Algorithmic Trading: Winning Strategies and Their Rationale*. John Wiley & Sons, May 2013. ISBN 978-1-118-46014-6. Google-Books-ID: WAIFDwAAQBAJ.

[31] J. Rudy, C. Dunis, G. Giorgioni, and J. Laws. Statistical Arbitrage and High-Frequency Data with an Application to Eurostoxx 50 Equities. *SSRN Electronic Journal*, 2010. ISSN 1556-5068. doi: 10.2139/ssrn.2272605.

[32] V. Holý and P. Tomanová. Estimation of Ornstein-Uhlenbeck Process Using Ultra-High-Frequency Data with Application to Intraday Pairs Trading Strategy. *arXiv:1811.09312 [q-fin]*, Dec. 2019. arXiv: 1811.09312.

[33] C.-F. Huang, C.-J. Hsu, C.-C. Chen, B. R. Chang, and C.-A. Li. An Intelligent Model for Pairs Trading Using Genetic Algorithms, 2015.

[34] M. Bayram, M. Akat, and S. Bulkan. Algorithmic pairs trading with expert inputs, a fuzzy statistical arbitrage framework. *Journal of Intelligent & Fuzzy Systems*, 38(1):697–707, Jan. 2020. ISSN 10641246, 18758967. doi: 10.3233/JIFS-179442.

[35] R. Bellman. On the Theory of Dynamic Programming. *Proceedings of the National Academy of Sciences*, 38(8):716–719, Aug. 1952. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.38.8.716.

[36] M. Minsky. Steps toward Artificial Intelligence. *Proceedings of the IRE*, 49(1):8–30, Jan. 1961. ISSN 0096-8390. doi: 10.1109/JRPROC.1961.287775.

[37] W. Mischel, E. B. Ebbesen, and A. Raskoff Zeiss. Cognitive and attentional mechanisms in delay of gratification. *Journal of Personality and Social Psychology*, 21(2):204–218, 1972. ISSN 1939-1315, 0022-3514. doi: 10.1037/h0032198.

[38] M. P. Deisenroth. A Survey on Policy Search for Robotics. *Foundations and Trends in Robotics*, 2 (1-2):1–142, 2011. ISSN 1935-8253, 1935-8261. doi: 10.1561/2300000021.

[39] J. Koutník, G. Cuccu, J. Schmidhuber, and F. Gomez. Evolving large-scale neural networks for vision-based reinforcement learning. In *Proceeding of the fifteenth annual conference on Genetic*

and evolutionary computation conference - GECCO '13, page 1061, Amsterdam, The Netherlands, 2013. ACM Press. ISBN 978-1-4503-1963-8. doi: 10.1145/2463372.2463509.

[40] rail.eecs.berkeley.edu. "Deep Reinforcement Learning: CS 285 Fall 2020". [Online]. Available: `http://rail.eecs.berkeley.edu/deeprlcourse/`. [Accessed: 20 October 2020].

[41] Y. Wang, D. Wang, S. Zhang, Y. Feng, S. Li, and Q. Zhou. Deep q-trading. 2017.

[42] G. Jeong and H. Y. Kim. Improving financial trading decisions using deep Q-learning: Predicting the number of shares, action strategies, and transfer learning. *Expert Systems with Applications*, 117:125–138, Mar. 2019. ISSN 0957-4174. doi: 10.1016/j.eswa.2018.09.036.

[43] P. Teetor. Better Hedge Ratios for Spread Trading. page 11, Nov. 2011.

[44] I. Gregory, C.-O. Ewald, and P. Knox. Analytical Pairs Trading Under Different Assumptions on the Spread and Ratio Dynamics. *SSRN Electronic Journal*, 2010. ISSN 1556-5068. doi: 10.2139/ssrn.1663703.

[45] J. Armstrong. *Principles of Forecasting: A Handbook for Researchers and Practitioners*. Kluwer Academic Publishers, Norwell, MA, 2001.

[46] J. Caldeira and G. V. Moura. Selection of a Portfolio of Pairs Based on Cointegration: A Statistical Arbitrage Strategy. *SSRN Electronic Journal*, 2013. ISSN 1556-5068. doi: 10.2139/ssrn.2196391.

[47] C. Krauss, X. A. Do, and N. Huck. Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European Journal of Operational Research*, 259(2):689–702, June 2017. ISSN 0377-2217. doi: 10.1016/j.ejor.2016.10.031.

[48] J. Ramos-Requena, J. Trinidad-Segovia, and M. Sánchez-Granero. Introducing Hurst exponent in pair trading. *Physica A: Statistical Mechanics and its Applications*, 488:39–45, Dec. 2017. ISSN 03784371. doi: 10.1016/j.physa.2017.06.032.

[49] E. Greensmith, P. L. Bartlett, and J. Baxter. Variance Reduction Techniques for Gradient Estimates in Reinforcement Learning. *Journal of Machine Learning Research*, page 60, 2004.

[50] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel. High-Dimensional Continuous Control Using Generalized Advantage Estimation. *arXiv:1506.02438 [cs]*, Oct. 2018. arXiv: 1506.02438.

[51] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Harley, T. P. Lillicrap, D. Silver, and K. Kavukcuoglu. Asynchronous Methods for Deep Reinforcement Learning. *International Conference on Machine Learning*, 48:10, 2016.

[52] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger. Deep Reinforcement Learning that Matters. *arXiv:1709.06560 [cs, stat]*, Jan. 2019. arXiv: 1709.06560.

[53] X. Glorot, A. Bordes, and Y. Bengio. Deep Sparse Rectifier Neural Networks. *AISTATS*, 15:315–323, 2011.

[54] J. Bergstra and Y. Bengio. Random Search for Hyper-Parameter Optimization. *The Journal of Machine Learning Research*, page 25, Feb. 2012.

[55] J. Engelberg, P. Gao, and R. Jagannathan. An Anatomy of Pairs Trading: The Role of Idiosyncratic News, Common Information and Liquidity. SSRN Scholarly Paper ID 1330689, Social Science Research Network, Rochester, NY, Nov. 2008.

[56] B. Do and R. Faff. Are Pairs Trading Profits Robust to Trading Costs? *Journal of Financial Research*, 35(2):261–287, 2012. ISSN 1475-6803. doi: 10.1111/j.1475-6803.2012.01317.x.

[57] R. T. Smith and X. Xu. A good pair: alternative pairs-trading strategies. *Financial Markets and Portfolio Management*, 31(1):1–26, Feb. 2017. ISSN 1934-4554, 2373-8529. doi: 10.1007/s11408-016-0280-x.

[58] G. Papadakis and P. Wysocki. *Pairs Trading and Accounting Information*. Sept. 2007.

[59] C. Alexander and A. Dimitriu. The Cointegration Alpha: Enhanced Index Tracking and Long-Short Equity Market Neutral Strategies. *SSRN Electronic Journal*, 2002. ISSN 1556-5068. doi: 10.2139/ssrn.315619.

[60] virtu.com. "Global Cost Review". [Online]. Available: `https://www.virtu.com/uploads/2019/02/ITG-Global-Cost-Review-4Q18.pdf`. [Accessed: 20 November 2020].

[61] W. F. Sharpe. The Sharpe Ratio. *The Journal of Portfolio Management*, 21(1):49–58, Oct. 1994. ISSN 0095-4918, 2168-8656. doi: 10.3905/jpm.1994.409501. Publisher: Institutional Investor Journals Umbrella Section: Primary Article.

[62] treasury.gov. "Daily Treasury Bill Rates Data". [Online]. Available: `https://www.treasury.gov/resource-center/data-chart-center/interest-rates/Pages/TextView.aspx?data=billrates`. [Accessed: 23 November 2020].

[63] A. W. Lo. The Statistics of Sharpe Ratios. *Financial Analysts Journal*, 58(4):36–52, July 2002. ISSN 0015-198X, 1938-3312. doi: 10.2469/faj.v58.n4.2453.

[64] C. Keating and W. Shadwick. A Universal Performance Measure. *Journal of Performance Measurement*, 6, Jan. 2002.

[65] P. N. Kolm and G. Ritter. Modern Perspectives on Reinforcement Learning in Finance. *SSRN Electronic Journal*, 2019. ISSN 1556-5068. doi: 10.2139/ssrn.3449401.

Appendix A

Bellman Equations

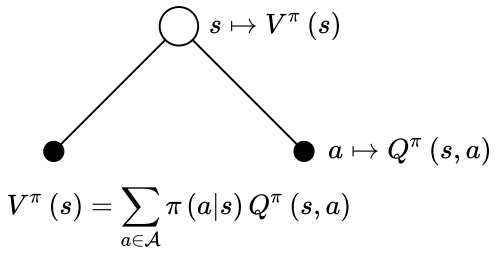
This appendix delves into the Bellman equations in order to give the reader some more intuitive insights. Let us recall the definition of an MDP and the associated terminology. An MDP is defined by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ where:

- $\mathcal{S}$ is a finite set of states;
- $\mathcal{A}$ is a finite set of actions;
- $\mathcal{P}$ is a state transition probability, $\mathcal{P}_{ss'}^{a} = \mathbb{P}\left[S_{t+1} = s' | S_t = s, A_t = a\right]$;
- $\mathcal{R}$ is a reward function, $\mathcal{R}_{s}^{a} = \mathbb{E}\left[R_{t+1} | S_t = s, A_t = a\right]$;
- γ is a discount factor where $\gamma \in [0, 1]$.

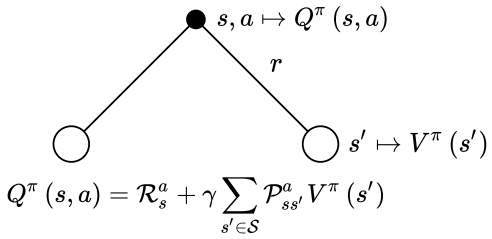
We will now analyze the Bellman equations, separating them in Bellman expectation equation and Bellman optimality equation.

A.1 Bellman Expectation Equation

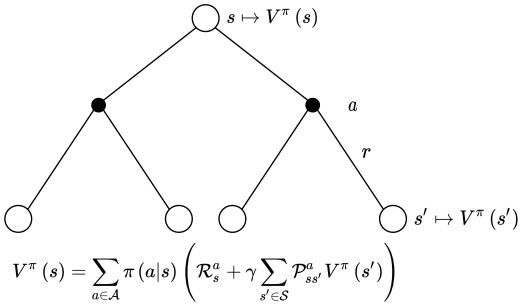
Figure A.1 shows the backup diagram for both the state-value function and the action-value function and the associated Bellman expectation equations respectively.



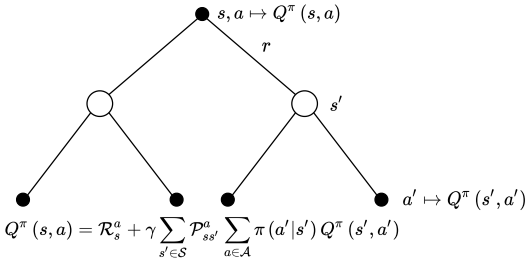
(a) Bellman Expectation Equation for V^{π}



(b) Bellman Expectation Equation for Q^{π}



(c) Bellman Expectation Equation for V^{π} (2)



(d) Bellman Expectation Equation for Q^{π} (2)

Figure A.1: Bellman Expectation Equation.

The backup diagram in figure A.1(a) describes the value of being in a particular state. From the state s, we take one of the two actions with some probability. There is a Q-value for each of the actions. Averaging the Q-values tell us how good it is to be in a particular state. The equation below the mentioned figure tells us the connection between the state-value function and the action-value function.

The backup diagram in figure A.1(b) says that suppose we start by taking some action a. So, taking into account action a, the agent might be blown to any of the states s' by the environment. Therefore, the question we might ask is how good it is to take action a? Again, we average the state-values of both states, added with an immediate reward which tells us how good it is to take a particular action a. The equation below the described figure is the definition of our Q-value.

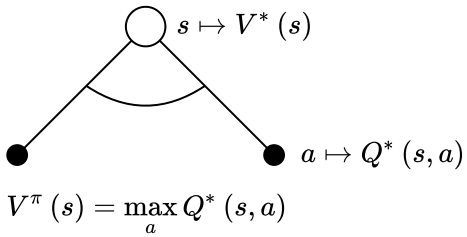
From the backup diagram in figure A.1(c), if our agent is in some state s and from that state suppose our agent can take two actions, our environment might take the agent to any of the states s'. We should note that our policy weights the probability of our agent's action from state s. After taking action a, the probability that we land in any of the states s' is weighted by the environment. The question now is: how good it is to be in state s after taking some action a, landing on another state s' and following our policy π after that? It is similar to what we have done before. We are going to average the value of successor states (s') with some transition probability ($\mathcal{P}$) weighted with our policy.

Figure A.1(d) is very similar to what we did in state-value function, being just the inverse. So this backup diagram says that our agent takes some action a, and the environment might take us to any of the states s. Then from that state, we can choose to take any actions a' weighted with the probability of our policy π. Again, we average them together, which gives us how good it is to take a particular action following a particular policy π all along.

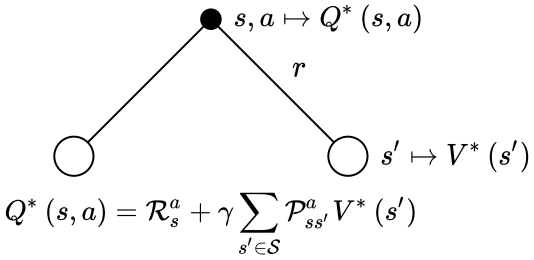
A.2 Bellman Optimality Equation

So we formulate the Bellman expectation equation for a given MDP to find both the state-value function and action-value function. Nevertheless, this does not tell us the best way to behave in an MDP. For that, we need the optimal value and the optimal policy function.

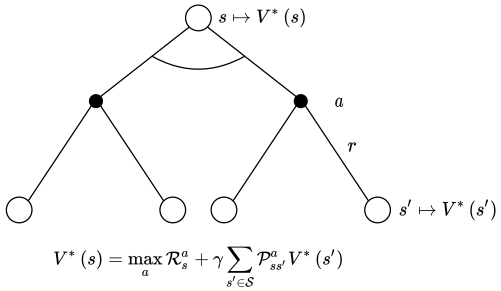
On the one hand, the optimal value function is the maximum value function over all policies. On the other hand, we say that a policy is an optimal policy if the value function with that policy is greater than the value function with any other policy for all states. However, how do we find this optimal policy? With the Bellman optimality equation. The optimal value function is recursively related to the Bellman optimality equation, and the figure A.2 illustrates the backup diagram of these equations.



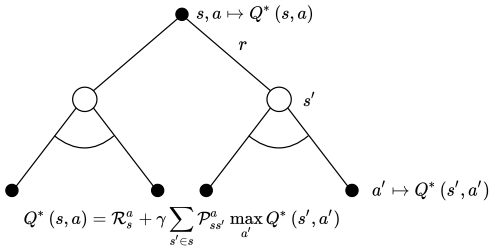
(a) Bellman Optimality Equation for V^π



(b) Bellman Optimality Equation for Q^π



(c) Bellman Optimality Equation for V^π (2)



(d) Bellman Optimality Equation for Q^π (2)

Figure A.2: Bellman Optimality Equation.

To not be redundant, all the reasoning presented above can be applied again; only this time, instead of taking the average, our agent takes the action with more value.

Appendix B

Analytical details of pair selection metrics

This appendix explores the analytical part of the calculation of the Hurst exponent value and half-life of mean-reversion, two of the metrics used in the framework implemented in the selection of pairs.

B.1 Hurst Exponent

A critical feature of a stationary price series is that the series diffuse from their initial value at a rate slower than that of a Geometric Brownian Motion. By measuring the rate of this diffusive behaviour, we can identify the nature of the time series.

The Hurst exponent calculation idea is to use the log price series variance to assess the diffusive behaviour rate. Formally, the variance is given by

$$\text{Var}\left(\tau\right) = \left\langle \left| z\left(t+\tau\right) - z\left(t\right) \right|^2 \right\rangle,$$

where $z\left(t\right)$ is the logarithmic time series value at time t, τ is an arbitrary time lag and $\langle \cdot \rangle$ is the average across time.

Since we are comparing the rate of diffusion to that of a Geometric Brownian Motion, we can use the fact that at large τ we have that the variance is proportional to τ in the case of a Geometric Brownian Motion,

$$\left\langle \left| z\left(t+\tau\right) - z\left(t\right) \right|^2 \right\rangle \sim \tau.$$

The critical insight is that if any autocorrelations exist (i.e. any sequential price movements possess non-zero correlation), then the above relationship is not valid. Instead, it can be modified to include an exponent value "$2H$", which gives us the Hurst Exponent value H,

$$\left\langle \left| z\left(t+\tau\right) - z\left(t\right) \right|^2 \right\rangle \sim \tau^{2H}.$$

Therefore, the Hurst Exponent can be characterized as an indicator of the degree of mean-reversion or trendiness. As H approaches zero, we are in the presence of a highly mean-reverting series, while for H near one the series is strongly trending.

B.2 Half-life of mean-reversion

The half-life of mean-reversion is a measure of how long it takes for a time series to mean-revert and can be calculated as an alternative interpretation of the lambda coefficient defined in section 2.1.2 [30]. So, the discrete-time series model described in equation 2.5 can be transformed into its differential form, so that the price changes now become infinitesimal quantities. Additionally, the term β may be ignored since we are dealing with a price series and the constant drift in price, if any, tends to be of much less magnitude than daily fluctuations. If the lagged difference terms are also ignored for further simplification, the expression can be written as

$$dy\left(t\right) = \left(\lambda y\left(t-1\right) + \mu\right) dt + d\epsilon.$$

This expression describes an Ornstein-Uhlenbeck stochastic process, where ϵ is Gaussian noise. This differential form leads to an analytical solution for the expected value of $y\left(t\right)$ as

$$\mathbb{E}\left(y\left(t\right)\right) = y_0\, e^{\lambda t} - \frac{\mu}{\lambda\left(1 - e^{\lambda t}\right)}.$$

Analyzing the above equation, we conclude, once again, that for $\lambda > 0$ the time series will not be mean-reverting, as stated in section 2.1.2. For a negative value of λ, the expected value of the time series decays exponentially to the value of $-\frac{\mu}{\lambda}$ with the half-life of decay equals to $-\frac{\log(2)}{\lambda}$. This result implies that the expected duration of mean-reversion is inversely proportional to the absolute value of λ, so the higher the absolute value of λ, the faster the time series revert to the mean.

Appendix C

List of ETFs

This appendix contains a list of the total universe of ETFs considered for trading in this work. The ticker symbol of each ETF is represented alongside the corresponding segment and category.

Table C.1: List of ETFs (1).

Ticker	Segment	Category	Ticker	Segment	Category
AAAU	Commodities: Precious Metals Gold	Precious Metals	DBO	Commodities: Energy Crude Oil	Energy
AGQ	Leveraged Commodities: Precious Metals Silver	Precious Metals	DBP	Commodities: Precious Metals	Precious Metals
AMJ	Equity: U.S. MLPs	Energy	DBS	Commodities: Precious Metals Silver	Precious Metals
AMJL	Leveraged Equity: U.S. MLPs	Energy	DDG	Inverse Equity: U.S. Energy	Energy
AMLP	Equity: U.S. MLPs	Energy	DGAZ	Inverse Commodities: Energy Natural Gas	Energy
AMU	Equity: U.S. MLPs	Energy	DGL	Commodities: Precious Metals Gold	Precious Metals
AMUB	Equity: U.S. MLPs	Energy	DGP	Leveraged Commodities: Precious Metals Gold	Precious Metals
AMZA	Equity: U.S. MLPs	Energy	DGZ	Inverse Commodities: Precious Metals Gold	Precious Metals
AOIL	Commodities: Energy	Energy	DIG	Leveraged Equity: U.S. Energy	Energy
ATMP	Equity: U.S. MLPs	Energy	DJCI	Commodities: Broad Market	Broad Market
BAL	Commodities: Agriculture Cotton	Agriculture	DJP	Commodities: Broad Market	Broad Market
BAR	Commodities: Precious Metals Gold	Precious Metals	DRIP	Inverse Equity: U.S. Oil & Gas Exploration & Production	Energy
BATT	Equity: Global Metals & Mining	Base Metals	DTO	Inverse Commodities: Energy Crude Oil	Energy
BCD	Commodities: Broad Market	Broad Market	DUG	Inverse Equity: U.S. Energy	Energy
BCI	Commodities: Broad Market	Broad Market	DUST	Inverse Equity: Global Gold Miners	Precious Metals
BCM	Commodities: Broad Market	Broad Market	DWT	Inverse Commodities: Energy Crude Oil	Energy
BMLP	Equity: U.S. MLPs	Energy	DZZ	Inverse Commodities: Precious Metals Gold	Precious Metals
BNO	Commodities: Energy Crude Oil	Energy	EMLP	Equity: U.S. MLPs	Energy
BOIL	Leveraged Commodities: Energy Natural Gas	Energy	ENFR	Equity: U.S. MLPs	Energy
CANE	Commodities: Agriculture Sugar	Agriculture	ERX	Leveraged Equity: U.S. Energy	Energy
CGW	Equity: Global Water	Agriculture	ERY	Inverse Equity: U.S. Energy	Energy
CHIE	Equity: China Energy	Energy	FCG	Equity: U.S. Natural Gas	Energy
CMDY	Commodities: Broad Market	Broad Market	FENY	Equity: U.S. Energy	Energy
COM	Commodities: Broad Market	Broad Market	FILL	Equity: Global Oil & Gas Exploration & Production	Energy
COMB	Commodities: Broad Market	Broad Market	FIW	Equity: Global Water	Agriculture
COMG	Commodities: Broad Market	Broad Market	FRAK	Equity: Global Oil & Gas	Energy
COPX	Equity: Global Metals & Mining	Base Metals	FTGC	Commodities: Broad Market	Broad Market
CORN	Commodities: Agriculture Corn	Agriculture	FTXN	Equity: U.S. Oil & Gas	Energy
COW	Commodities: Agriculture Livestock	Agriculture	FUD	Commodities: Agriculture	Agriculture
CPER	Commodities: Industrial Metals Copper	Base Metals	FUE	Commodities: Agriculture Grains	Agriculture
CRAK	Equity: Global Oil & Gas	Energy	FXN	Equity: U.S. Energy	Energy
DBA	Commodities: Agriculture	Agriculture	GASL	Leveraged Equity: U.S. Natural Gas	Energy
DBB	Commodities: Industrial Metals	Base Metals	GASX	Inverse Equity: U.S. Natural Gas	Energy
DBC	Commodities: Broad Market	Broad Market	GAZ	Commodities: Energy Natural Gas	Energy
DBE	Commodities: Energy	Energy	GCC	Commodities: Broad Market	Broad Market

Table C.2: List of ETFs (2).

Ticker	Segment	Category	Ticker	Segment	Category
GDX	Equity: Global Gold Miners	Precious Metals	JJU	Commodities: Industrial Metals Aluminum	Base Metals
GDXJ	Equity: Global Gold Miners	Precious Metals	JNUG	Leveraged Equity: Global Gold Miners	Precious Metals
GDXS	Inverse Equity: Global Gold Miners	Precious Metals	JO	Commodities: Agriculture Coffee	Agriculture
GDXX	Leveraged Equity: Global Gold Miners	Precious Metals	KOL	Equity: Global Coal	Energy
GLD	Commodities: Precious Metals Gold	Precious Metals	KOLD	Inverse Commodities: Energy Natural Gas	Energy
GLDI	Commodities: Precious Metals Gold	Precious Metals	LD	Commodities: Industrial Metals Lead	Base Metals
GLDM	Commodities: Precious Metals Gold	Precious Metals	LIT	Equity: Global Metals & Mining	Base Metals
GLDW	Commodities: Precious Metals Gold	Precious Metals	MLPA	Equity: U.S. MLPs	Energy
GLL	Inverse Commodities: Precious Metals Gold	Precious Metals	MLPB	Equity: U.S. MLPs	Energy
GLTR	Commodities: Precious Metals	Precious Metals	MLPC	Equity: U.S. MLPs	Energy
GOAU	Equity: Global Gold Miners	Precious Metals	MLPE	Equity: U.S. MLPs	Energy
GOEX	Equity: Global Gold Miners	Precious Metals	MLPG	Equity: U.S. MLPs	Energy
GRU	Commodities: Agriculture Grains	Agriculture	MLPI	Equity: U.S. MLPs	Energy
GSC	Commodities: Broad Market	Broad Market	MLPO	Equity: U.S. MLPs	Energy
GSG	Commodities: Broad Market	Broad Market	MLPQ	Leveraged Equity: U.S. MLPs	Energy
GSP	Commodities: Broad Market	Broad Market	MLPX	Equity: U.S. MLPs	Energy
GUSH	Leveraged Equity: U.S. Oil & Gas Exploration & Production	Energy	MLPY	Equity: U.S. MLPs	Energy
IAU	Commodities: Precious Metals Gold	Precious Metals	MLPZ	Leveraged Equity: U.S. MLPs	Energy
IAUF	Commodities: Precious Metals Gold	Precious Metals	NIB	Commodities: Agriculture Cocoa	Agriculture
IEO	Equity: U.S. Oil & Gas Exploration & Production	Energy	NLR	Equity: Global Nuclear Energy	Energy
IEZ	Equity: U.S. Oil & Gas Equipment & Services	Energy	NRGD	Inverse Equity: U.S. Oil & Gas	Energy
IMLP	Equity: U.S. MLPs	Energy	NRGO	Leveraged Equity: U.S. Oil & Gas	Energy
IXC	Equity: Global Energy	Energy	NRGU	Leveraged Equity: U.S. Oil & Gas	Energy
IYE	Equity: U.S. Energy	Energy	NRGZ	Inverse Equity: U.S. Oil & Gas	Energy
JDST	Inverse Equity: Global Gold Miners	Precious Metals	NUGT	Leveraged Equity: Global Gold Miners	Precious Metals
JHME	Equity: U.S. Energy	Energy	OIH	Equity: Global Oil & Gas Equipment & Services	Energy
JJA	Commodities: Agriculture	Agriculture	OIL	Commodities: Energy Crude Oil	Energy
JJC	Commodities: Industrial Metals Copper	Base Metals	OILD	Inverse Commodities: Energy Crude Oil	Energy
JJE	Commodities: Energy	Energy	OILK	Commodities: Energy Crude Oil	Energy
JJG	Commodities: Agriculture Grains	Agriculture	OILU	Leveraged Commodities: Energy Crude Oil	Energy
JJM	Commodities: Industrial Metals	Base Metals	OILX	Commodities: Energy Crude Oil	Energy
JJN	Commodities: Industrial Metals Nickel	Base Metals	OUNZ	Commodities: Precious Metals Gold	Precious Metals
JJP	Commodities: Precious Metals	Precious Metals	PALL	Commodities: Precious Metals Palladium	Precious Metals
JJS	Commodities: Agriculture Softs	Agriculture	PDBC	Commodities: Broad Market	Broad Market
JJT	Commodities: Industrial Metals Tin	Base Metals	PGM	Commodities: Precious Metals Platinum	Precious Metals

Table C.3: List of ETFs (3).

Ticker	Segment	Category	Ticker	Segment	Category
PHO	Equity: Global Water	Agriculture	UAG	Commodities: Agriculture	Agriculture
PICK	Equity: Global Metals & Mining	Base Metals	UBG	Commodities: Precious Metals Gold	Precious Metals
PIO	Equity: Global Water	Agriculture	UCI	Commodities: Broad Market	Broad Market
PLTM	Commodities: Precious Metals Platinum	Precious Metals	UCIB	Commodities: Broad Market	Broad Market
PPLN	Equity: U.S. MLPs	Energy	UCO	Leveraged Commodities: Energy Crude Oil	Energy
PPLT	Commodities: Precious Metals Platinum	Precious Metals	UGA	Commodities: Energy Gasoline	Energy
PSCE	Equity: U.S. Energy	Energy	UGAZ	Leveraged Commodities: Energy Natural Gas	Energy
PXE	Equity: U.S. Oil & Gas Exploration & Production	Energy	UGL	Leveraged Commodities: Precious Metals Gold	Precious Metals
PXI	Equity: U.S. Energy	Energy	UNG	Commodities: Energy Natural Gas	Energy
PXJ	Equity: U.S. Oil & Gas Equipment & Services	Energy	UNL	Commodities: Energy Natural Gas	Energy
PYPE	Equity: U.S. Energy	Energy	URA	Equity: Global Nuclear Energy	Base Metals
REMX	Equity: Global Metals & Mining	Base Metals	USAI	Equity: U.S. MLPs	Energy
RING	Equity: Global Gold Miners	Precious Metals	USCI	Commodities: Broad Market	Broad Market
RJA	Commodities: Agriculture	Agriculture	USL	Commodities: Energy Crude Oil	Energy
RJI	Commodities: Broad Market	Broad Market	USO	Commodities: Energy Crude Oil	Energy
RJN	Commodities: Energy	Energy	USOD	Inverse Commodities: Energy Crude Oil	Energy
RJZ	Commodities: Broad Market Metals	Broad Market	USOI	Commodities: Energy Crude Oil	Energy
RYE	Equity: U.S. Energy	Energy	USOU	Leveraged Commodities: Energy Crude Oil	Energy
SCO	Inverse Commodities: Energy Crude Oil	Energy	USV	Commodities: Precious Metals Silver	Precious Metals
SDCI	Commodities: Broad Market	Broad Market	UWT	Leveraged Commodities: Energy Crude Oil	Energy
SGDJ	Equity: Global Gold Miners	Precious Metals	VDE	Equity: U.S. Energy	Energy
SGDM	Equity: Global Gold Miners	Precious Metals	WEAT	Commodities: Agriculture Wheat	Agriculture
SGG	Commodities: Agriculture Sugar	Agriculture	WTID	Inverse Commodities: Energy Crude Oil	Energy
SGOL	Commodities: Precious Metals Gold	Precious Metals	WTIU	Leveraged Commodities: Energy Crude Oil	Energy
SIL	Equity: Global Silver Miners	Precious Metals	XES	Equity: U.S. Oil & Gas Equipment & Services	Energy
SILJ	Equity: Global Silver Miners	Precious Metals	XLE	Equity: U.S. Energy	Energy
SIVR	Commodities: Precious Metals Silver	Precious Metals	XLEY	Equity: U.S. Energy	Energy
SLV	Commodities: Precious Metals Silver	Precious Metals	XME	Equity: U.S. Metals & Mining	Base Metals
SLVO	Commodities: Precious Metals Silver	Precious Metals	XOP	Equity: U.S. Oil & Gas Exploration & Production	Energy
SLVP	Equity: Global Silver Miners	Precious Metals	YGRN	Inverse Equity: U.S. Oil & Gas	Energy
SOYB	Commodities: Agriculture Soybeans	Agriculture	YMLI	Equity: U.S. MLPs	Energy
TAGS	Commodities: Agriculture	Agriculture	YMLP	Equity: U.S. MLPs	Energy
TBLU	Equity: Global Water	Agriculture	ZMLP	Equity: U.S. MLPs	Energy
TPYP	Equity: U.S. MLPs	Energy	ZSL	Inverse Commodities: Precious Metals Silver	Precious Metals

Appendix D

Sharpe Ratio Scale Factors

This appendix contains the table describing the Sharpe ratio scale factors proposed by Lo [63] and adopted in this work. This scale factor depend on the return's autocorrelation value ρ, and the aggregation value q, as illustrated in table D.1. In this study, q is set to 250, since daily data corresponding to one year is being aggregated. The parameter ρ is measured according to each portfolio returns.

Table D.1: Scale factors for time-aggregated Sharpe ratios when returns follow an AR(1) process for various aggregation values and first-order autocorrelations. Source: [63].

ρ (%)	Aggregation Value, q									
	2	3	4	6	12	24	36	48	125	250
90	1.03	1.05	1.07	1.10	1.21	1.41	1.60	1.77	2.67	3.70
80	1.05	1.10	1.14	1.21	1.43	1.81	2.14	2.42	3.79	5.32
70	1.08	1.15	1.21	1.33	1.65	2.19	2.62	3.00	4.75	6.68
60	1.12	1.21	1.30	1.46	1.89	2.55	3.08	3.53	5.63	7.94
50	1.15	1.28	1.39	1.60	2.12	2.91	3.53	4.06	6.49	9.15
40	1.20	1.35	1.49	1.75	2.36	3.27	3.98	4.58	7.35	10.37
30	1.24	1.43	1.60	1.91	2.61	3.65	4.44	5.12	8.23	11.62
20	1.29	1.52	1.73	2.07	2.88	4.04	4.93	5.68	9.14	12.92
10	1.35	1.62	1.86	2.25	3.16	4.45	5.44	6.28	10.12	14.31
0	**1.41**	**1.73**	**2.00**	**2.45**	**3.46**	**4.90**	**6.00**	**6.93**	**11.18**	**15.81**
–10	1.49	1.85	2.16	2.66	3.80	5.39	6.61	7.64	12.35	17.47
–20	1.58	1.99	2.33	2.90	4.17	5.95	7.31	8.45	13.67	19.35
–30	1.69	2.13	2.53	3.17	4.60	6.59	8.10	9.38	15.20	21.52
–40	1.83	2.29	2.75	3.48	5.09	7.34	9.05	10.48	17.01	24.11
–50	2.00	2.45	3.02	3.84	5.69	8.26	10.21	11.84	19.26	27.31
–60	2.24	2.61	3.37	4.30	6.44	9.44	11.70	13.59	22.19	31.50
–70	2.58	2.76	3.86	4.92	7.45	11.05	13.77	16.04	26.33	37.43
–80	3.16	2.89	4.66	5.91	8.96	13.50	16.98	19.88	32.96	47.02
–90	4.47	2.97	6.47	8.09	12.06	18.29	23.32	27.61	46.99	67.65

Note that the row corresponding to $\rho = 0\%$ is the IDD case in which the scale factor is simply $\sqrt{q}$ (e.g., $\sqrt{250} = 15.81$). And that for each holding-period q, positive serial correlation reduces the scale

factor below the IID value and negative serial correlation increases it, as described in section 4.4.2.

Appendix E

Trading Perfomance

This appendix includes a table of the five trials' average trading performance for each pair for the DRL model compared to the standard threshold-based model in the trading period of 2018. The trading period of 2019 is considered for analysis in section 5.3.

Table E.1: Comparison of trading performance per pair for the trading period of 2018.

Trading Model		**Standard Trading Model**			**DRL Trading Model**		
Evaluation Metrics		**ROI**	**SR**	**MDD**	**ROI**	**SR**	**MDD**
Pairs	**UCO / DBE**	22.96%	1.23	-15.97%	11.50%	0.40	-32.06%
	DBE / USO	2.70%	0.22	-4.81%	9.26%	0.37	-28.35%
	DBO / USL	11.03%	0.55	-22.44%	-10.04%	-0.67	-27.89%
	DBO / USO	15.67%	0.51	-33.52%	14.14%	0.44	-34.78%
	DBP / DBS	6.12%	0.80	-3.58%	-5.74%	-0.67	-9.65%
	DGL / DBP	-1.99%	-1.15	-2.55%	-2.71%	-1.35	-3.24%
	DGP / DBP	0%	-	0%	-1.94%	-0.42	-6.03%
	DBP / GDX	6.47%	0.79	-4.93%	-1.94%	-0.42	-6.03%
	DBP / SLV	-0.15%	-0.18	-6.80%	5.35%	0.49	-5.56%
	UGL / DBP	0%	-	0%	-1.26%	-0.80	-2.13%