Advantage Actor-Critic Algorithm Application to Pairs Trading Strategy

Diogo Rodrigues
diogoegrodrigues@tecnico.ulisboa.pt

Nuno Horta
nuno.horta@tecnico.ulisboa.pt

***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.**

***Index Terms*—Pairs Trading, Market Neutral, Total Least Squares, Reinforcement Learning, Deep Learning, Actor-Critic**

I. Introduction

Pairs trading is an important long/short equity investment tool widely used by hedge funds and institutional investors for decades. It is well-known by enabling traders to virtually profit from any market direction.

The classic form of pairs trading is based on finding two securities with some relation 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. Accordingly, it short sells the other one and expects the pair's prices to converge within the intended time horizon.

In the early days, pairs trading methods were quite popular due to the opportunities to obtain arbitrage profit. However, with so many investors sought these arbitrage opportunities, its profitability began to deteriorate. Recently, significant research has been conducted to overcome these shortcomings in the strategy [1], [2].

In this work, we address two studies: (i) how two key features (extract trading signals with different methods and vary the window sizes) impact the profitability of the strategy, and (ii) if it is possible to train a deep reinforcement learning (DRL) model to optimise the pairs trading strategy beyond the traditional way.

We choose this type of learning because, recently, it has become one of the most active research areas that has made it very much cutting-edge. The most significant achievements were introduced by a company called DeepMind. In 2013, its first pioneering paper [3] stunned the artificial intelligence (AI) community with a computer program based on the reinforcement learning (RL) that had taught itself to play seven different Atari video games, three of them at human expert level. It was a remarkable result because it only used pixel positions and game scores as input, and made no adjustment of the architecture or learning algorithm between games. By 2015 the system achieved superhuman performance in over 20 different Atari games [4].

The rest of this document is organised as follows. Section II introduces the main concepts of pairs trading while describing the related work. Besides, it presents the RL basis. Section III illustrates the proposed model for studying the two topics highlighted above. Section IV includes some practical information on how the investigation was conducted. Section V shows the results and provides a discussion of the experiments. Finally, section VI produces some concluding remarks.

II. Background and Related Work

Each stage composing pairs trading strategy is described in detail below, together with the RL paradigm foundations. For each subject, the most relevant related work is presented.

A. Pairs Selection

The investor must define what criteria should be used to select a pair. Reference [5] provides a comprehensive survey of the literature, indicating that the most common approaches are the distance and cointegration approaches.

The most cited paper in pairs trading and the most prominent study for distance-based selection criteria is [6]. The authors selected the pairs of all possible combinations that generated the minimum sum of Euclidean squared distance (SSD) between the two securities' price series. Although widely used, [5] noted that this metric is analytically sub-optimal. If $p_{i,t}$ denote a realization of the normalised price process $P_i = (P_{i,t})_{t \in T}$ of a security i, the average sum of

squared distances $\overline{ssd}_{P_i,P_j}$ in the formation period[1] of a pair formed by securities i and j is given by

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

The ideal pair, according to the SSD criterion, is the one that minimises (1). However, this would mean a spread of zero, making not consistent with the idea of potentially profitable pairs. If there are no deviations of any kind, there will be no trade opportunities. A revision and analyse of the distance approach is made in [7], finding diminishing profitability in recent years.

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,

$$S_t = Y_t - \beta X_t, \tag{2}$$

where β is the cointegration factor[2], 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. The most cited work in this field is [8] that proposes a set of heuristics for cointegration-based strategies. Furthermore, [9] performed a comparison study between the two approaches and showed that selected pairs based on cointegration more often exhibit mean-reverting behaviour and are more profitable than distance pairs. The main reason is that cointegration identifies econometrically more sound equilibrium relationships.

B. Trading Strategy

The traditional framework for trading execution is the one proposed by [6], and can be described as below:

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.
3) Convert the spread into a Z-score[3] 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.

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.

[1]The formation period corresponds to the period used to find the most appealing candidate pairs.

[2]In pairs trading, the cointegration factor is common called hedge ratio.

[3]In statistics, a Z-score is the number of standard deviations by which the value of a raw score is above or below the mean value.

C. Reinforcement Learning

RL's essence is to learn by interacting and receiving feedback in the form of a reward signal. 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. 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.

RL basically solves the problem defined by Markov decision process (MDP). It consists of a 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 matrix, $\mathcal{R}$ is a reward function, and γ is a discount factor. The main goal is to find a policy π that maximises the expected sum of discounted future rewards an agent will receive after step t,

$$G_t = R_{t+1} + \gamma R_{t+2} + \ldots = \sum_{k=0}^{\infty}\gamma^k R_{t+k+1}. \tag{3}$$

There are two main types of RL methods: value-based that try to find how good it was to reach a particular state or take a specific action, and policy-based algorithms that try to directly find the optimal policy. In DRL, we deal with parameterised policies, whose outputs are computable functions (artificial neural networks), which we can adjust to change the behaviour via gradient descent.

In this study, we optimise the pairs trading strategy with a type of game using advantage actor-critic algorithm, a combination of the two described approaches. 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). 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. This idea is represented in Fig. 1.

Considering policy π_θ, using gradient descent to maximise performance, the policy weights θ are updated in the direction of

$$\mathop{\mathbb{E}}_{\tau\sim\pi_\theta}\left[\sum_{t=1}^{H}\nabla_\theta \log \pi_\theta\left(a_t \mid s_t\right)\sum_{t'=t}^{H}\gamma^{t'-t}R\left(s_{t'}, a_{t'}\right)\right], \tag{4}$$

where τ is a trajectory (sequence of states and actions), s_t is the state at time t, and a_t is the action at time t. Equation (4) represents the traditional policy gradient methods using Monte-Carlo (MC) updates. However, these suffer from high variance and low convergence.

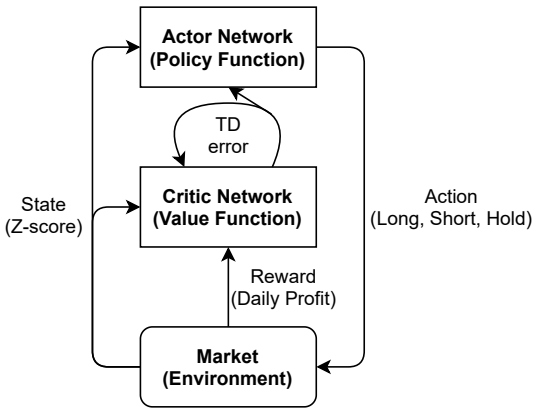


Fig. 1. The state-action-reward cycle for the actor-critic framework. Adapted from: [10].

It is possible to reduce variance and increase stability by manipulating the cumulative reward term and subtracting a baseline. There are multiple ways to do this [11]. We decided to use what is called in literature the advantage function, defined as the temporal-difference (TD) error,

$$A^{\pi}\left(s_t, a_t\right) = R_{t+1} + \gamma V^{\pi}\left(s_{t+1}\right) - V^{\pi}\left(s_t\right), \tag{5}$$

where V^{π} is the expected return starting from the given state and always act according to policy π. We make the critic learn the advantage values, and so, the evaluation of an action is based not only on how good it was but also on how good it can be. The main advantage is being able to reduce policy networks variance and stabilise the model.

III. Proposed Model

The proposed model's flow in a simplistic way can be represented just like Fig. 2. Initially, after the dataset 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. Here, we apply the advantage actor-critic algorithm.

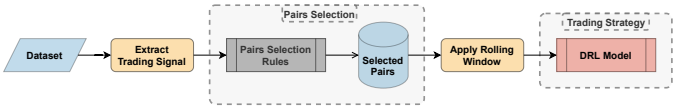


Fig. 2. Proposed model architecture.

Each component of the proposed model is explored below. We will start by presenting the alternative method for the trading signal extraction, then moving through the framework for selecting pairs and concluding with a detailed exposition of the DRL model. We decided to leave the formalities about the rolling window block to the next section because it is a more practical consideration.

A. Trading Signal

The extraction of the signal for trading is a fundamental procedure for both stages in pairs trading strategy that in most of the literature, it is neglected.

In regression analysis, the ordinary least squares (OLS) yields predictions of a dependent variable contingent on an independent variable and minimises the sum of squared errors of prediction. Assume that x_i, y_i, and ε_i are an independent variable, a dependent variable, and an error term. We can estimate β from the following equation by taking a partial derivative:

$$y_i = \beta x_i + \varepsilon_i \tag{6}$$

$$\sum_{i=1}^{n} \left(y_i - \beta x_i\right)^2 \tag{7}$$

$$\beta = \left(\sum_{i=1}^{n} x_i' \, x_i\right)^{-1} \sum_{i=1}^{n} x_i' \, y_i. \tag{8}$$

OLS seems to be the workhorse among all the literature on cointegration-based pairs trading. However, it assumes that the independent variable is an observed score known without error, and all of the error is connected to the dependent variable. This assumption implies that by swapping the dependent and independent variable, the hedge ratios will not be symmetrical, making this method inconsistent.

We propose to study a possible better approach of total least squares (TLS) along with [12]. 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, the observed values of X_i and Y_i have the following error terms:

$$X_i = x_i + u_i \qquad\qquad Y_i = y_i + e_i \tag{9}$$

where x_i and y_i are the true scores 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 X and Y and the error variance ratio can be represented as

$$s_X^2 = s_x^2 + s_u^2 \qquad s_Y^2 = s_y^2 + s_e^2 \qquad \delta = \frac{s_e^2}{s_u^2} \tag{10}$$

It is assumed that the true score combination between X and Y is linear,

$$y_i = \beta_0 + \beta_1 x_i. \tag{11}$$

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

$$\beta_1 = \frac{s_Y^2 - \delta s_X^2 + \left[\left(s_Y^2 - \delta s_X^2\right)^2 + 4\delta s_{YX}^2\right]^{\frac{1}{2}}}{2 s_{YX}}, \tag{12}$$

where s_{YX} is the covariance between Y and X. Fig. 3 provides a more visual intuition on both methods for better understanding.

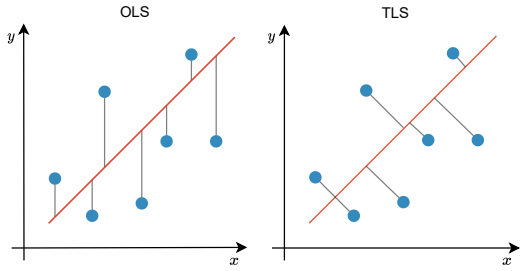


Fig. 3. Visual comparison between OLS and TLS.

To sum things up, cointegration analysis using OLS will be sensitive to the ordering of variables. One of the relationships may be cointegrated, while the other will not. We would expect that both choices will yield the same conclusion if the variables are truly cointegrated. So it is worth including the TLS approach in this work. To mitigate OLS's issue, we propose to run the Engle-Granger test[4] 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. The epsilon value (residual cointegration) is used as a trading signal through Z-scoring.

B. Pair Selection Framework

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 and searching for possible pairs. Here, two methodologies can be found in the literature:

- Search for pairs without restrictions, resulting in a higher computational cost and leading to the multiple comparison problem[5], 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.

[4]The Engle-Granger test is widely used to test for cointegration.

[5]The multiple comparison problem is an increased chance of classifying hypothesis tests incorrectly when running many tests.

Concerning point (ii), and following the line of reasoning presented above, we decided to follow the set of criteria presented by [1] to guarantee a more robust pairs selection framework. This set of criteria is a unification of methods applied in separate research works. First, 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 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. If the Hurst exponent value, H, is less than 0.5, it indicates a mean-reverting time series. Therefore we require this condition to be met. Third, a mean-reverting spread alone does not generate profits. There must be consistency between the mean-reversion duration and the trading period. The half-life metric can be interpreted as an estimate of the expected time for the spread's mean-reversion [13], and this step is used to filter pairs whose half-life is not consistent with the trading period (less than one day or more than one year). Finally, it is imposed that the spread crosses the mean at least 12 times per year to provide sufficient liquidity, which ideally will be equivalent to a minimum of one cross per month, on average.

C. Trading Strategy

The pairs trading strategy is considered a kind of a game, where we can profit if a position is opened and closed in the right times. Therefore, we will implement a system that tries to optimise trading decisions given a spread. 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 action space is made up of just three action signals due to our action space's discretisation. 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 I.

TABLE I
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

The agent interacts with the market simulation to create an environment to coordinate information flow that reaches

the system to follow the RL paradigm. 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. When it receives an action, the market simulation updates all intrinsic information and drafts a new state and a scalar reward for the chosen action.

The state consists of the next set of input features for both neural networks. We used delta prices to define the states, as it has proven empirically to be better than 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 to focus on relative values and try to find opportunities that bridge the gaps in the standard threshold-based model.

The reward signal is defined as 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. Even so, defining the reward signal only in this way often led the system to get stuck in a local optimum, 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 aiming to penalise large variations of unrealised profit which are interpreted as a risk. In essence, our reward system is very similar to one of these alternatives, the Sortino ratio, since positive volatility is not punished, as is 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.

An additional note pointed out by [14] is that in gradient-based methods, a large and sparse output scale can result in problems regarding saturation and learning inefficiency. During training, the rewards are standardised to overcome this problem, a technique known as reward clipping, which compresses the space of estimated expected returns.

It is essential to accentuate that the loss function regarding artificial neural networks cannot be interpreted in the same way as supervised learning. The standard loss function is commonly defined on a fixed data distribution independent of the parameters we aim to optimise. Not valid in RL, where the data must be sampled on the current policy. The other main difference is that it does not measure performance. 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.

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

$$L_{actor} = -\sum_{t=1}^{T} \log \pi_\theta \left(a_t \mid s_t\right) A^{\pi}_{\theta_v} \left(s_t, a_t\right), \qquad (13)$$

where T is the number of timesteps per episode, s_t is the state at timestep t, a_t chosen action at timestep t, 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 (13) 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, which corresponds to the spread of action probabilities.

The critic loss function is simpler, being nothing more than the difference of our estimated return and the value function. So, training the critic can be set up as a regression problem with the following loss function:

$$L_{critic} = L_\delta \left(R_{t+1} + \gamma V^{\pi}_{\theta_v} \left(s_{t+1}\right),\ V^{\pi}_{\theta_v} \left(s_t\right)\right), \qquad (14)$$

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

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. L2 regularisation and dropout are applied as regularisation techniques.

IV. Test Planning and Validation

It is time to present the dataset used as well as describe some of the data processing conducted. Besides, we will raise the most critical considerations in the implementation of the study design. It should be noted that testing the proposed model as Fig. 2 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 investigate the impact of each study being addressed. Finally, the characteristics of the trading simulation and evaluation metrics used are also highlighted.

A. Dataset

ETFs are the security of choice in our work. ETFs could provide some risk-minimising elements that cannot be obtained when using single stocks. In its nature, an ETFs tries to replicate the return on an index consisting of multiple securities, achieving diversification benefits as it is exposed to a basket of assets.

We have fixed our dataset to a group composed only by commodity-linked ETFs. This decision not only reduces the number of pairs at the outset, making everything computationally faster but also allows a more careful analysis. This subset of ETFs is the dataset of choice in the work of [1].

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.

A total of 208 commodity-linked ETFs were available for trading in January 2020. By choosing just ETFs active through an entire period; we are aware that survivorship bias[6] is introduced. To try to limit the impact of this bias, the most recent possible periods were considered.

Having selected ETFs' universe for our study, we collected the adjusted daily closing prices from January 2, 2009, to December 31, 2019. Those ETFs with missing values throughout the period being considered are removed. Then, 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.[7] Following the criterion adopted in [6], [15], we use trading volume to filter out ETFs that have at least one day without trading. Finally, some sporadic outliers are corrected manually.

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 cross-validation procedure does not fit financial data applications. It can permit peeking into the future (look-ahead bias), leading to unrealistically optimistic results. To avoid this problem, we propose using role forward cross-validation.

The periods for simulating each study phase are exhibited in Fig. 4. 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 study phase 2, a 7-year long formation period is proposed. Here, more formation data is required to fit the DRL model.

B. Study Phase 1

There seems to be no consensus in the literature on the best window size to use. Reference [6] initially presented a 12-month formation period followed by a 6-month trading period. References [9], [16] have conducted empirical tests on the two periods' lengths. Both conclude that 12-month formation period provides superior results. In contrast, [2] 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. The window size

[6]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.

[7]Very briefly, trading illiquid ETFs would result in a higher bid-ask spread, which would significantly impact profit margins.

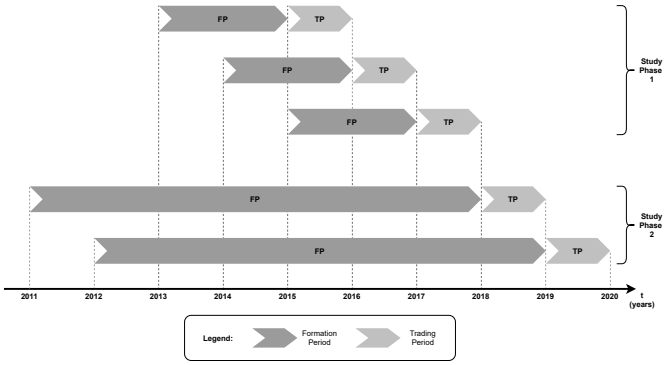


Fig. 4. Data partition periods.

decision will depend on the type of strategy to adopt and the dataset in question, 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 II displays the various window sizes considered both in the number of months and in the respective number of days.[8]

TABLE II
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 (Fig. 4). 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. This procedure allows a finer recalibration of the trading signal parameters representing the pair's constituents' current relationship more accurately. Fig. 5 visually illustrates how

[8]The average number of trading days in 1 year is about 252 days.

this process is done for the example where we consider a 4-month formation window and a 2-month trading window.

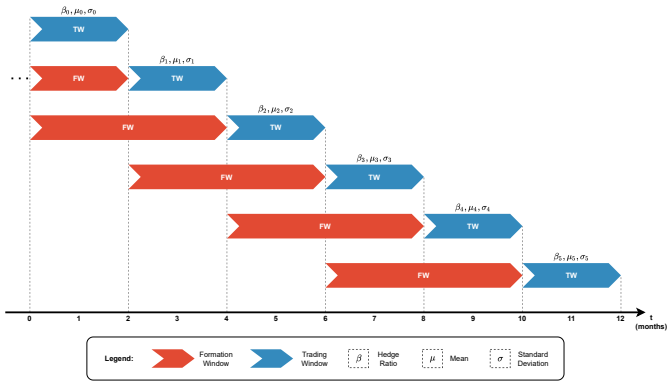


Fig. 5. Rolling window scheme.

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

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. We apply the standard threshold-based trading model proposed in [6], with the parameters specified in Table III.

TABLE III
THRESHOLD-BASED TRADING MODEL PARAMETERS

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

C. Study Phase 2

This study phase aims to compare the robustness provided by the standard threshold-based trading model with the proposed DRL model. Comparing 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 in extracting 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.

The DRL 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 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.

D. Trading Simulation

Concerning the portfolio construction, we impose 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.

Next, we still need to define how we are going to allocate the capital for each pair. We assume that the capital earned by the short position could be used to cover the long position. Most hedge funds adopt this type of leverage, and it is a so-called self-financing strategy. On this basis, we assume an initial investment of one dollar in each pair. With this approach, the return obtained by the pair can be interpreted directly. To deal with the hedge ratio, we have decided to follow the criteria of [1], [17] and respect the hedge ratio between the pair's constituents, as illustrated in Fig. 6. As the trading progresses, we consider that all the capital earned in a trade position is reinvested in the next trade.

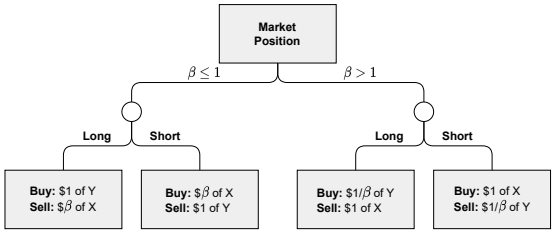


Fig. 6. Market position definition. Adapted from: [1].

This work did not consider the one-day delay rule, adopted by [6], [16].[9] Instead, we decide to pay the entire bid-ask spread as a transaction cost. This choice is also made because waiting one day can also misrepresent our DRL model's decisions' profitability.

All the results presented in this work account for transaction costs. The transaction costs considered are based on estimates from [15], in which the authors perform an in-depth study on the impact of transaction costs in pairs trading. The costs comprise three components: commission (8 bps), bid-ask spread[10] (20 bps) and short-selling (1% per annum).

E. Evaluation Metrics

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

The ROI is calculated as the net return on an investment divided by the investment cost, which we enforced to be one dollar.

The annual portfolio SR is computed as

[9]The one-day delay is an attempt to mitigate the bid-ask bounce.

[10]Reference [15] refers to the bid-ask spread cost as the market impact cost.

$$SR_{\text{year}} = \frac{R^{\text{port}} - R_f}{\sigma_{\text{port}}} \times \text{annualisation factor}, \quad (15)$$

where R^{port} represents the expected daily portfolio returns and R_f the risk-free rate[11]. The portfolio volatility, σ_{port}, is given by

$$\sigma_{\text{port}} = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N} \omega_i \omega_j \text{Cov}_{i,j}}, \quad (16)$$

where ω_i represent the weight of the i-th pair in a portfolio of size N. The annualisation factor is set according to the correction factor proposed in [19], to prevent imprecise approximations.

The MDD is calculated as

$$MDD\,(T) = \max_{\tau \in (0,\,T)} \left[\max_{t \in (0,\,\tau)} \frac{P\,(t) - P\,(\tau)}{P\,(t)} \times 100\% \right]. \quad (17)$$

V. Results

The results obtained for each study phase are presented next.

A. Study Phase 1

We start by detailing the pairs selection procedure in Table IV.

TABLE IV
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	
Discard 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

The first thing that becomes clear is the confirmation that the cointegration test has a profound impact considering that most filtering occurs in this stage. 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). 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, with a slight tendency for TLS to select fewer pairs. In practical terms, this behaviour is not unfamiliar; the hedge ratios obtained by OLS and TLS do not differ much, but when they do differ, that difference is likely to be significant.

Before we go into the trading results, we want to give the reader a taste of how the six different window sizes change the spread. Fig. 7 illustrates exactly this, exemplifying with an arbitrary pair and period since similar behaviour can be found for all pairs and periods under study.

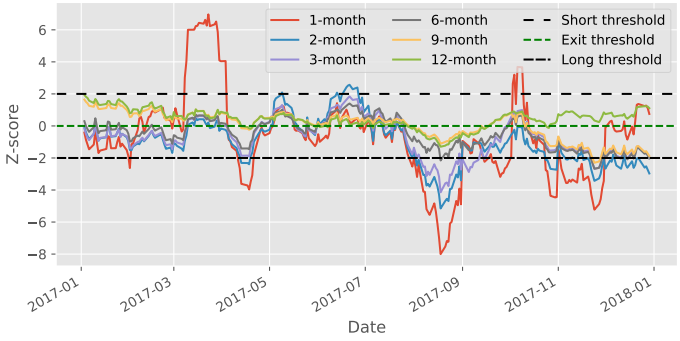


Fig. 7. Comparison of applying the six different trading window sizes to the pair of ETFs, DBO and PXE, using TLS and in the trading period of 2017.

The signals seem quite different but, with a closer look, we notice that the variations are similar, in the sense that when one is going up, the other is also going up. In 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.

Concerning the trading performance, Table V unveils the results regarding the unseen data. 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.

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 the consideration of realistic transaction costs as reported by [15]. 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.

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 (with one exception), the trading signals made with TLS method are better than those made with OLS method. This superiority of TLS is based on the difference between the hedge ratios of the two methods and confirm all the evidence, proving that TLS best captures the functional relationship establishing the pair.

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 and look at it from a consistency perspective, the 12-month window is more reliable, as it achieves positive profits on average for both linear regressions. It is clear the importance and influence

[11]This study follows the most common practice of using the interest paid on a three-month U.S. Treasury bill, taken from [18], during the corresponding test period and converted into daily returns to be consistent with (15).

TABLE V
TRADING PERFORMANCE 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%

of the trading window length on trading performance, not being a minor factor in optimising the pairs trading strategy parameters.

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. Table VI 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.

TABLE VI
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 ± 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 ± 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 ± 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 ± 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 ± 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 ± 2	39%	41%	59%	60%	31%	30%	43%	44%

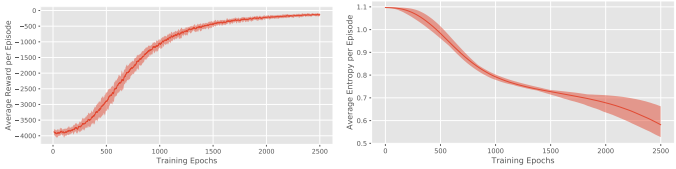
We can find a correlation between a high percentage of observations outside the thresholds for opening positions and poor trading performance. If we add a larger magnitude of the mean deviation of the distribution and the theoretical mean, this correlation becomes even more evident. This interdependence is especially true in 2016 when performance is worse, corroborated by a high percentage value (with one exception). We can confirm that this higher percentage is synonymous of more divergent pairs.

B. Study Phase 2

Based on the information collected throughout the study phase 1, our options for this phase are to adopt the TLS to extract the trading signal, limit the top 10 pairs, and use the 12-month trading window. Before we get into the trading performance and testing, it is crucial to confirm that our DRL algorithm is trained well. In DRL, the only way to determine how training is going on is through the average rewards collected for each episode. In addition to rewards, plotting the entropy provides valuable information about how well the model is performing. The model has three outputs (one for each action), and if each action is equally probable, then entropy is roughly 1.1.[12] As long as the logged entropy keeps this value, the model does not perform better than randomly picking an action.

Reference [14] 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, and following the authors' suggestion, our results are an average of five trials with different random seeds to attest our results' reproducibility.

Presenting the training process for all ten pairs for each test period was inconceivable, so we decided to exhibit one arbitrary pair. Fig. 8 illustrate on the left the development of the average sum of rewards per episode and the right the average entropy per episode. The solid line represents the five trials' average and the filled region between the maximum and minimum values for rewards and entropy trials.



(a) Rewards learning curve. (b) Entropy learning curve.

Fig. 8. Learning curves for the pair of ETFs, UCO and DBE, in the training period from Jan 2011 to Dec 2017.

We can find that the average sum of rewards per episode steadily increased, indicating that the DRL model is appropriately trained. In the entropy plot, we can see that it starts just below 1.1, and then it starts to drop after a while.

[12]Recall that the Shannon entropy for this example is defined by $-3 \cdot \frac{1}{3} \cdot \ln \frac{1}{3}$.

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 VII shows the average performance measures for each pair in the trading period of 2019.

TABLE VII
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.

Turning now the analysis to the portfolios and adding the trading period of 2018, the average performance measures obtained are shown in Table VIII.

TABLE VIII
TRADING PERFORMANCE 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.

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. 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.

VI. CONCLUSIONS

We divide the study of pairs trading in two phases. First, we proposed using the TLS method to extract the trading signal, and the results showed that for all six window sizes and periods studied performed better. Additionally, we opened a door for how vital the window size is in the strategy's definition, given its impact on the results. Secondly, we designed and implemented an advantage actor-critic agent to make pairs trading decisions. We confirmed that the model was well trained and that during test periods outperforms the traditional pairs trading strategy on average for both out-of-sample datasets.

REFERENCES

[1] S. M. Sarmento and N. Horta, "Enhancing a Pairs Trading strategy with the application of Machine Learning," *Expert Systems with Applications*, vol. 158, p. 113490, Nov. 2020.

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

[3] 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.

[4] 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*, vol. 518, no. 7540, pp. 529–533, Feb. 2015, number: 7540 Publisher: Nature Publishing Group.

[5] C. Krauss, "Statistical Arbitrage Pairs Trading Strategies: Review and Outlook," *Journal of Economic Surveys*, vol. 31, no. 2, pp. 513–545, 2017.

[6] E. Gatev, W. N. Goetzmann, and K. G. Rouwenhorst, "Pairs Trading: Performance of a Relative-Value Arbitrage Rule," *The Review of Financial Studies*, vol. 19, no. 3, pp. 797–827, Oct. 2006.

[7] B. Do and R. Faff, "Does Simple Pairs Trading Still Work?" *Financial Analysts Journal*, vol. 66, no. 4, pp. 83–95, Jul. 2010.

[8] G. Vidyamurthy, *Pairs Trading: Quantitative Methods and Analysis*. John Wiley & Sons, Aug. 2004, google-Books-ID: FTFuPFx0hdcC.

[9] N. Huck and K. Afawubo, "Pairs trading and selection methods: is cointegration superior?" *Applied Economics*, vol. 47, no. 6, pp. 599–613, Feb. 2015.

[10] R. S. Sutton and A. G. Barto, "Reinforcement Learning: An Introduction," p. 352, 1998.

[11] 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.

[12] I. Gregory, C.-O. Ewald, and P. Knox, "Analytical Pairs Trading Under Different Assumptions on the Spread and Ratio Dynamics," *SSRN Electronic Journal*, 2010.

[13] E. Chan, *Algorithmic Trading: Winning Strategies and Their Rationale*. John Wiley & Sons, May 2013, google-Books-ID: WAlFDwAAQBAJ.

[14] 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.

[15] B. Do and R. Faff, "Are Pairs Trading Profits Robust to Trading Costs?" *Journal of Financial Research*, vol. 35, no. 2, pp. 261–287, 2012.

[16] R. T. Smith and X. Xu, "A good pair: alternative pairs-trading strategies," *Financial Markets and Portfolio Management*, vol. 31, no. 1, pp. 1–26, Feb. 2017.

[17] H. Rad, R. K. Y. Low, and R. Faff, "The profitability of pairs trading strategies: distance, cointegration and copula methods," *Quantitative Finance*, vol. 16, no. 10, pp. 1541–1558, Oct. 2016.

[18] 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].

[19] A. W. Lo, "The Statistics of Sharpe Ratios," *Financial Analysts Journal*, vol. 58, no. 4, pp. 36–52, Jul. 2002.