Deep Reinforcement Learning Applied to Statistical Arbitrage Investment Strategy on Cryptomarket

Abstract

In the complex and dynamic nature of financial markets where increasingly sophisticated investment strategies are required, deep reinforcement learning (DRL) has proven successful in generating real-time investment strategies, outperforming classical models. Alongside this, statistical arbitrage strategies exploit temporary market inefficiencies to generate returns. In this regard, a novel DRL-based arbitrage method has been developed. This paper presents a unified framework that combines classical statistical arbitrage theory with deep reinforcement learning (DRL) techniques to generate investment strategies. The framework addresses the challenges of identifying similar asset portfolios, extracting signals indicating temporary price deviations, and determining optimal trading rules given market conditions. The proposed methodology involves constructing arbitrage portfolios based on cointegration relationships, removing signals from price series and portfolios, and using a DRL agent to make optimal decisions within a fixed time horizon. The empirical analysis focuses on the cryptocurrency market, known for its volatility and risk. Results demonstrate that DRL agents can generate strategies with positive returns in out-of-sample periods, outperforming market benchmarks. Moreover, these strategies effectively reduce risk, achieving higher risk-adjusted returns on individual assets. The strategies maintain positive returns when considering transaction costs, with the DRL agent outperforming the standard arbitrage strategy. Their randomness and coherence are studied to verify the robustness of the agents' decisions. The actions generated by the agents are not random but based on well-founded policies, which align with the obtained results.

Keywords: statistical arbitrage, pairs trading, deep reinforcement learning, cointegration, investment strategies, cryptocurrency market

1. Introduction

Gatev et al. (2006) [1] introduced the concept of pairs trading, a simple process that follows two guidelines. Firstly, two assets are sought whose prices have a joint movement throughout history in a formation period. Secondly, the price difference or spread between the assets is monitored during a trading period. If the prices diverge and the spread widens, the loser is bought, and the winner is sold. In the case of selected assets or stocks with an equilibrium relationship, the spread is expected to revert to its historical mean. Thus, positions are reversed, and profit can be obtained from this. Subsequently, a natural modification to pairs trading is to extend it to a set of assets, rather than pairs, giving rise to statistical arbitrage. Statistical arbitrage encompasses a variety of investment strategies that identify and exploit temporary price differences between similar assets using statistical tools, which is understood as profiting from market inefficiencies. While its most basic form corresponds to pairs trading, these strategies do not bet on the market direction but on temporary differences or misvaluations during short periods in assets.

Considering that every arbitrage strategy requires solving three fundamental problems: Given a large number of assets, which are the portfolios of similar assets that represent a joint movement? Given these portfolios, what are the signals indicating the presence of a temporary price deviation? Finally, given these signals, how should these assets be traded to optimize profitability given market conditions? Firstly, obtaining "similar" assets is ambiguous, and finding portfolios with long and short positions is difficult. Secondly, what information should be considered to determine patterns and the complex relationships that allow identifying these market inefficiencies? Given these complex market relationships, the optimal trading rule maximizes the investor's objective. These challenges require flexible tools that utilize all available information. Naturally, deep learning techniques have proven highly efficient in solving problems in high-dimensional spaces and finding complex relationships between variables. However, the issue at hand does not correspond to a typical problem of point prediction and categorization, but rather a comprehensive framework considering a horizon and a dynamic decision rule.

Considering these background aspects, this process is well represented as an online decision-making process as market conditions change. According to the requirements of the arbitrage scheme, deep reinforcement learning (DRL) functions as a direct adaptive optimal control system. The agent in DRL is the component that decides which actions to take to maximize the established reward. Therefore, the fundamental principle of DRL is to maximize the cumulative reward of the agent in the learning process, which leads to optimal decision-making strategies for different types of problems. Reinforcement learning has proven successful in various tasks, ranging from unmanned driving [2] and robot navigation [3] to playing video games. It has achieved the proposed tasks and, in some cases, surpassed human experts, such as AlphaZero defeating world champions in Shogi, Chess, and Go [4]. Hence, the question arises in the field of quantitative trading: Is it possible to train a DRL agent that can generate arbitrage strategies, obtain profits and reduce risk in a highly volatile and risky market like cryptocurrency?

This work proposes a unified framework that combines elements from classical statistical arbitrage theory and innovates in applying DRL to generate investment strategies. Firstly, there is a module for generating arbitrage portfolios through cointegration. Then, signal extraction is performed based on the generated prices and portfolios. Finally, the DRL module takes these signals, processes them, and based on the experience gained during training, makes optimal decisions to maximize the accumulated return. A fundamental difference is that

the actions are not generated to maximize each individual action but within a given time horizon. Since the cointegration relationships among assets do not persist for long periods, they must be dynamically generated and used only for a short period.

A statistical perspective is used to construct the portfolios based on the cointegration relationship of price series. In this way, two portfolios of different assets replicating each other are built, creating market neutrality by taking opposing positions in each portfolio. Therefore, the difference in the value of these synthetic assets captures the temporary deviations in the prices of the underlying assets. Subsequently, various transformations and classic technical indicators are employed to detect patterns and obtain relevant information about the market state, considering aspects such as asset valuation, trend, volatility, and momentum. Thus, the market state is decomposed into different signals, which serve as input to the DRL agent or, in other words, represent the market state observed by the agent. Finally, to generate the optimal investment rule, the agent processes this market information using a neural network and makes decisions based on it to maximize the overall return of a trading session. Since each session involves a different arbitrage portfolio, generating a simulator of random scenarios on which the agent can be trained is necessary. For this purpose, a broad historical window is used, encompassing periods of extreme trend and volatility and less volatile periods with no clear trend. The proposed scenarios correspond to past events, not synthetic data, as generating synthetic market data is not feasible. In this sense, training the agent involves maximizing the accumulated reward over a finite horizon, much like maximizing the score in a video game where random scenarios are generated.

The training was conducted using a training window from 2020-11-01 to 2022-10-11, comprising 34,000 records with a frequency of 30 minutes for 14 assets. Subsequently, the out-of-sample analysis was performed from 2022-10-19 to 2023-03-18, totaling 150 days of transactions. The results show, firstly, that DRL agents can generate strategies with positive returns in the out-of-sample period. These findings are particularly interesting as they demonstrate the success of an arbitrage strategy in an extremely risky and trending market, such as cryptocurrencies.

Secondly, the employed strategies significantly reduce risk, generating low-risk returns compared to the market. Furthermore, when considering different risk-adjusted performance metrics at daily and intraday trading levels, they achieve higher returns than the best-performing assets while maintaining lower risk levels.

Thirdly, when introducing transaction costs, the strategies maintain positive returns except for the A2C agent, which performs worst. However, the DQN agent not only outperforms the standard COIN arbitrage strategy without transaction costs but also improves upon it when considering these costs, while maintaining lower risk levels than the market.

Lastly, it is determined that the actions the agents generate are not random. Still, the trained policies generate well-founded actions, not erratic ones, consistent with the results. Thus, it is established that generating an arbitrage strategy using DRL is possible, improving upon classical statistical methods.

The following structure of this paper is as follows: Section 2 presents a comprehensive investigation of related works. Section 3 introduces the formulated arbitrage model. First, define the arbitrage portfolio construction. Second, describe the market signal representation. Third, describing the DRL agent or arbitrageur, and finally, the training method. Section 4 analyzes the results obtained in different aspects, conducting a daily and intraday analysis of strategy risk along with risk-adjusted performance, and finally studying the agent's behavior. Lastly, Section 5 concludes the work.

2. Related Work

Recently, machine learning, in particular deep learning, has become an increasingly popular tool and technique for various applications in finance, including portfolio selection problems. The increasing availability of data and growing computer processing power have contributed to the popularity of deep learning. RL is a branch of machine learning based on behaviorist psychology. The purpose of RL is to learn how to achieve a goal, such as automated control or sequential decision-making by interacting with the environment through an adaptive learning process. Neuneier (1997) [5] formulated financial markets as MDPs under several assumptions and simplifications regarding market characteristics. Such formulation is crucial because finite MDPs enable Q-learning, and RL techniques, to converge on optimal policy. Furthermore, the development of DRL techniques enables the automatic learning of more sophisticated behaviors. DRL can achieve an outstanding performance of various realistic tasks through automatic feature engineering and end-to-end learning through gradient descent [6].

For the latest financial applications, Deng et al. (2016) [7] introduced a trading system incorporating a deep direct RL framework with a fuzzy representation of past returns, which greatly reduced market uncertainties, increasing the feasibility of extracting more complex features. A portfolio management framework was designed by using the EIIE topology [8], which was used to reveal the specific characteristic of individual assets without making biased decisions affected by any long-past poor records of particular assets. Neves et al. (2018)[9] generated a short-term speculation system in the foreign exchange market based on reinforcement learning. Liang et al.(2018)[10] implement three state-of-the-art RL algorithms for portfolio management: DDPF, PPO, and PG. They also propose an adversarial training method and reveal that it can significantly improve training efficiency. Lee et al. (2020) [11] propose the MultiAgent RL-based portfolio management System (MAPS) and devise a new loss function with a diversification penalty term to effectively encourage agents to maximize both diversity and returns. Wu et al. (2020)[12] introduced an adaptive trading method based on DRL using the GDQN (Gated Deep Q-learning) and GDPG (Gated Deterministic Policy Gradient) approaches, showing positive results. Zhang et al. (2020)[13] compared DRL in various markets using a reward function that incorporates a target conditional volatility level, achieving results that outperform classical methods and maintaining profits under high transaction costs. Liu et al. (2021)[14] proposed an automatic high-frequency transaction framework based on DRL using an LSTM as the policy based on a PPO algorithm on bitcoin. They compare standard prediction models, including SVM, MLP, LSTM, TCN, and Transformer, and conclude that LSTM is the best fit to use as the policy function. The empirical studies confirm that the proposed method performs Superior to various common trading strategy benchmark for a single financial product. Chen et al.(2021) [15] proposed a multimodal reinforcement trading system based on sentiment analysis which includes a deep neural network and multimodal deep recurrent neural network where the multimodal integration of price information with news sentiment enables the agent to make profits. Lin et al. (2022)[16] proposed a novel Multiagent-based DRL for risk-shifting portfolio management (MABDRL) with outstanding performance on several common metrics with a 37% Sharpe ratio, where each agent is equipped with sophisticated deep policy networks that enable the proposed RL agent to learn risk-shifting behaviors with stable convergence.

On the other hand, applications of DRL in arbitrage or pairs trading are rather scarce. Mulvey et al. (2020)[17] and Kim and Kim (2019)[18] employed machine learning methods in parametric statistical arbitrage models. In contrast, Pelger et al. (2021)[19] generalized the arbitrage model by incorporating principal compo-

nent analysis, instrumented principal component analysis (PCA and IPCA), and alpha factors models. They combined these arbitrage portfolios with convolutional networks, transformers, Fourier analysis, feed-forward networks, and the parametric Ornstein-Uhlenbeck model with thresholds for optimal investment decision generation.

3. Model Formulation

3.1. Arbitrage Portfolios

Considering a set of normalized prices $P_{t,n}$ of assetss $n = 1, \ldots, N$ at time $t = 1, \ldots T$. To generate the arbitrage portfolios we assume a Error Correction Model (ECM) for the log prices where $\ln(P_{n,t}) = p_{n,t}$ denoted as $\mathbf{p}_t = (p_{i,t}, \ldots p_{n,t})'$. Considering a $VAR(k)$ model on $\mathbf{p}_t$ which can be rewritten as:

$$\Delta \mathbf{p}_t = \mu + \Sigma_{i=1}^{k-1} \Gamma_i \Delta \mathbf{p}_{t-1} + \Pi \mathbf{p}_{t-1} + \varepsilon_t \tag{1}$$

where $\Gamma_i \in \mathbb{R}^{n \times n}$, $\Pi \in \mathbb{R}^{n \times n}$ and $\varepsilon_t \overset{iid}{\sim} N(0, \Lambda)$. If $rank(\Pi) = n$, then $\mathbf{p}_t$ is stationary. If $rank(\Pi) = 0$ then $\Pi = 0$ implying that $\Delta \mathbf{p}_t$ is a $VAR(k-1)$ process and there are no cointegration vectors found. If $1 \leq rank(\Pi) = r \leq n-1$, then there exist $n \times r$ matrices with rank r , $\mathbf{A}$ and $\mathbf{B}$, such that Π in model (1) can be expressed as:

$$\Pi = \mathbf{A}\mathbf{B}' \tag{2}$$

and $\mathbf{b}_1' \mathbf{p}_t, \ldots, \mathbf{b}_r' \mathbf{p}_t$ are stationary, where $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_r)$. The Model (1) with Π satisfying $\Pi = \mathbf{A}\mathbf{B}'$ is the ECM, with cointegration vectors $\mathbf{b}_1, \ldots, \mathbf{b}_r$. Then, from the $\mathbf{p}_t$ as the cumulative total log return of each asset, the r columns of $\mathbf{B}$ can be used to form r cointegrated (COIN) portfolios. However, these portfolios are not necessarily market neutral.

To generate the arbitrage portfolio, each of the cointegration vectors is normalized so that they make a dollar-neutral portfolio. Given the cointegration vector $\mathbf{b}_i = (b_i^1, \ldots, b_i^n)'$ defining the long and short positions for every asset:

$$k \in L_i \iff b_i^k \geq 0, \quad \forall i = 1, \ldots r \tag{3}$$

$$k \in S_i \iff b_i^k < 0, \quad \forall i = 1, \ldots r \tag{4}$$

then, the sets L_i and S_i define the directional position amount of each asset of the cointegrated portfolio formed by the cointegration vector $\mathbf{b}_i$.

The long portfolio is given by the assets where the cointegration coefficient is positive and the short portfolios by the negative coefficients. Therefore r dollar neutral portfolios are built. For each vector $\mathbf{b}_i$, the corresponding normalization constant are $l_i = \sum_{k \in L_i} |b_i^k|$ and $s_i = \sum_{k \in S_i} |b_i^k|$. So the weight of each asset in the portfolio is given by :

$$W_i^{(k)} = \begin{cases} b_i^{(k)} / l_i & \text{if } b_i^{(k)} \geq 0 \\ b_i^{(k)} / s_i & \text{if } b_i^{(k)} < 0 \end{cases} \tag{5}$$

These r portfolios are dollar-neutral, and the arbitrage portfolio $\mathbf{P}$ is built by a simple average of all the portfolios:

$$\mathbf{P} = \frac{1}{r} \sum_{k=1}^{r} \mathbf{W}_k, \quad \mathbf{W}_k = (W_k^1, \ldots, W_k^n) \tag{6}$$

Finally, the portfolio formed by the positive weights will be denoted as portfolio $\mathbf{A}$, and the portfolio formed by negative weights will be denoted portfolio $\mathbf{B}$ so that $\mathbf{P} = \mathbf{A} - \mathbf{B}$.

3.2. Signal Generation

With the corresponding arbitrage portfolio defined, the next step is to define the signals extracted based on market information. These signals are the input for the trading policy; in other words, they define the state of the market observed by the agent (the algorithm).

3.2.1. OLHC

Considering the weights of the portfolios, we have 30-minute candlestick data for each asset in the interval. Thus, the *Open*, *High*, *Low*, *Close*, and *Volume* values are constructed for assets A and B. The values are normalized based on the closing price of the first candle, which is set to 1, so the *Open*, *High*, and *Low* values of the first candle do not necessarily start at 1.

Considering these values, initially, eight states are defined, considering the normalized values for *OHLC*, $\mathbf{O}_t = (O_t^1, \ldots, O_t^N)'$, and similarly for the others:

$$\mathrm{O}_t^i = \mathrm{Open}_t^i / \mathrm{Close}_0^i, \forall i \tag{7}$$

$$\mathrm{H}_t^i = \mathrm{High}_t^i / \mathrm{Close}_0^i, \forall i \tag{8}$$

$$\mathrm{L}_t^i = \mathrm{Low}_t^i / \mathrm{Close}_0^i, \forall i \tag{9}$$

$$\mathrm{C}_t^i = \mathrm{Close}_t^i / \mathrm{Close}_0^i, \forall i \tag{10}$$

Considering the weights $W_A = (w_A^1, \ldots, w_A^N)'$ and $W_B = (w_B^1, \ldots, w_B^N)'$, where $\mathbf{P} = W_A - W_B$, we obtain the *OHLC* values for portfolios A and B:

$$O_t^A = \mathbf{O}_t W_A', \quad O_t^B = \mathbf{O}_t W_B' \tag{11}$$

$$H_t^A = \mathbf{H}_t W_A', \quad H_t^B = \mathbf{H}_t W_B' \tag{12}$$

$$L_t^A = \mathbf{O}_t W_A', \quad L_t^B = \mathbf{L}_t W_B' \tag{13}$$

$$C_t^A = \mathbf{C}_t W_A', \quad C_t^B = \mathbf{C}_t W_B' \tag{14}$$

$$\tag{15}$$

These series are used to construct other indicators; however, only the closing series is considered a state signal.

3.2.2. Moving Average Convergence Divergence

The Moving Average Convergence Divergence (MACD) is a popular technical indicator commonly used in stock trading to identify momentum, trend, and price strength changes. The MACD is calculated by subtracting the 26-period exponential moving average (EMA) from the 12-period EMA and then plotting a 9-period EMA of the MACD as a signal line.

$$\mathrm{MACD}(l, s)_t = \mathrm{EMA}(l)_t - \mathrm{EMA}(s)_t \tag{16}$$

where

$$\mathrm{EMA}(k)_t = \mathrm{Price}_t k + \mathrm{EMA}_{t-1}(1 - k) \tag{17}$$

This indicator is used with a long-term average of 14 periods and a short-term average of 5 periods. It will be calculated for the closing prices of portfolios A and B.

3.2.3. Relative Strength Index

The Relative Strength Index (RSI(N)) is defined as:

$RSI = 100 - 100/(1 + RS)$ where $RS = AvgU/AvgD$

$AvgU$ = Average of all up movements in N periods

$$AvgU = \frac{1}{N}\sum[\mathbf{Price}_{t-i} - \mathbf{Price}_{t-i-1}]^{+}$$

$AvgD$ = Average of all down movements in N periods

$$AvgD = \frac{1}{N}\sum[\mathbf{Price}_{t-i} - \mathbf{Price}_{t-i-1}]^{-}$$

This indicator is used with $N = 20$ based on the closing price. It is calculated for portfolios A and B, resulting in 2 additional series.

3.2.4. Dynamic Moving Average

A Kalman filter generates a moving average signal of the price for synthetic assets A and B. This filter models the price series P_t by considering a not observable moving average μ_t in the following way:

$$\text{Measurement Equation: } P_t = \mu_t + \varepsilon_t, \quad \varepsilon_t \sim N(0, \sigma^2) \tag{18}$$

$$\text{State Equation: } \mu_t = \mu_{t-1} + \nu_t, \quad \epsilon_t \sim N(0, \sigma_\mu^2) \tag{19}$$

Additionally, the Kalman gain is obtained:

$$K_t = \frac{\mathbb{C}\text{ov}\left[\mu_t, \nu_t|\mathcal{F}_{t-1}\right]}{\mathbb{V}\left[\nu_t|\mathcal{F}_{t-1}\right]} = \frac{\mathbb{C}\text{ov}\left[\mu_t, \nu_t|\mathcal{F}_{t-1}\right]}{\mathbb{V}\left[\mu_t|\mathcal{F}_{t-1}\right] + \mathbb{V}\left[\varepsilon_t|\mathcal{F}_{t-1}\right]}, \quad \nu_t = P_t - \mathbb{E}\left[P_t|\mathcal{F}_{t-1}\right] \tag{20}$$

Then, the conditional expectation of μ_t given the price P_t is:

$$\hat{\mu}_t = \mathbb{E}\left[\mu_t|P_t\right] = (1 - K_t)\mathbb{E}\left[\mu_t|\mathcal{F}_{t-1}\right] + K_t P_t \tag{21}$$

Where the volatilities are estimated in the training sample, more details about filters can be found in Durbin & Koopman 2012 [20].

3.2.5. Stock Strength

The *Stock Strength* indicator proposed in Wu, Chen, Wang, Troiano, Loia, and Fujita 2020 [12] aims to discriminate whether the market is in a trending or volatile regime:

$$ST = \frac{P_n - P_1}{\sum_{i=1}^{n}|P_{i+1} - P_1|} \tag{22}$$

It is applied to a window of 10 intervals for synthetic assets A and B. This indicator measures the accumulated return throughout n periods divided by the sum of the absolute returns for the period. Therefore, if the market is volatile, this indicator will be close to zero, while in a trending regime, it will be either strongly positive or strongly negative.

3.2.6. Exponentially Weighted Moving Average

The conditional volatility is calculated using the standard EWMA for the return series of synthetic assets A and B, along with the spread series A-B, to introduce dependence on conditional volatility.

$$\sigma_t^2 = \lambda\sigma_t^2 + (1 - \lambda)r_t^2 \tag{23}$$

A memory parameter $\lambda = 0.94$ is used, which means that a volatility shock has a high persistence over time.

3.2.7. Bias

The Bias measures the relative deviation of the price from a reference moving average value. In this case, it is applied to the moving average obtained through the Kalman filter.

$$\text{Bias}_t = \frac{P_t - \mu_t}{\mu_t} \tag{24}$$

This measure is applied to synthetic assets A and B.

3.2.8. Signs

Another measure of relative strength is the sign of the returns. Unlike the previous measures, it does not measure the net effect or quantitative strength comparison based on returns. Instead, it counts the number of positive returns versus negative returns.

$$\text{Signs}_t = \frac{\sum_{i=0}^{N} [\text{sign}(r_{t-i})]^+}{N} \tag{25}$$

This measure is applied to synthetic assets A and B, using a window of 10 periods. Unlike other measures, it takes a discrete set of values.

3.2.9. Stochastic Relative Strength Index

The stochastic RSI is also applied, which is a scaled version of the regular RSI using a scaler based on the minimum and maximum values over a certain period.

$$\text{SRSI}_t = \frac{\text{RSI}_t - \min_{k=t-N:t} \text{RSI}_k}{\max_{k=t-N:t} \text{RSI}_k - \min_{k=t-N:t} \text{RSI}_k} \tag{26}$$

This is applied to synthetic assets A and B using a window of 20 periods.

3.2.10. Candle Shadows

Finally, a metric of price volatility concerning the range of candlesticks in the period is applied.

$$\text{Upper}_t = \frac{\text{High}_t - \max\{\text{Open}_t, \text{Close}_t\}}{\text{High}_t - \text{Low}_t} \tag{27}$$

$$\text{Lower}_t = \frac{\min\{\text{Open}_t, \text{Close}_t\} - \text{Low}_t}{\text{High}_t - \text{Low}_t} \tag{28}$$

These measures are applied to the synthetic assets A and B.

Additionally, spread A - B is introduced as a state. The moving average calculated through the Kalman filter for the spread is also used as a state and, finally, a market turbulence index, which corresponds to the Mahalanobis distance of market returns:

$$\textbf{Turbulence: } T_t = \sqrt{(\mathbf{r}_t - \mu)' \Sigma^{-1} (\mathbf{r}_t - \mu)} \tag{29}$$

where μ corresponds to the vector of mean returns during the training period, r_t represents the returns of the assets, and Σ is the covariance matrix of the returns during the training period. This brings the total number of states to 26, which is expected to provide the agent with information about market interactions for learning purposes.

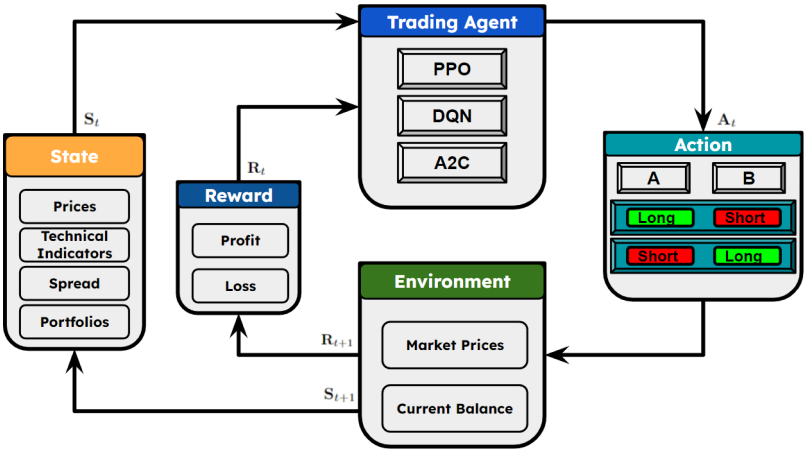

Figure 1: Interaction scheme of the deep reinforcement learning system for the trading application. The states correspond to those defined through technical indicators, volatilities, and others generated from synthetic assets and spreads. The possible actions are either taking a long position in A and a short position in B or taking a short position in A and a long position in B.

3.3. Trading Decision

3.3.1. Markov Decision Process

The trading problem can be formulated as a Markov Decision Process (MDP) where an agent interacts with the environment at discrete time steps. At every moment t, the agent observes the environment state S_t. Given the observed state, S_t, the agent chooses an action A_t, and based on the action, it gets a reward R_{t+1} on the next time step, and the agent observes a new state S_{t+1}. The interaction between the agent and the environment produces a trajectory of state, actions, and rewards $\tau = (S_0, A_0, R_1, S_1, A_1, R_2, \dots)$. The objective of the agent is to maximize the expected return at any given time denoted as G_t:

$$G_t = \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k \tag{30}$$

where γ is the discount factor.

The state is represented as the signals extracted from the arbitrage portfolios and spreads. The reward function is the discounted total cumulative reward with a discount factor of 0.995. The action space is discrete, as the portfolios are premade; the agent does not influence the portfolio composition; it only has to decide on which side to take. Given the two arbitrage portfolios **A** and **B**, the actions are long-**A** and short-**B** ($A_t = 1$) or short-**A** and long-**B** ($A_t = -1$), so there are only two possible actions to take, $A_t \in \{-1, 1\}$. Given the portfolio returns $r_t^A = \ln P_t^A - \ln P_{t-1}^A$ and $r_t^B = \ln P_t^B - \ln P_{t-1}^B$, the reward or return of the action A_{t-1} is given by

$$R_t = A_{t-1}\left[e^{r_t^A} - e^{r_t^B}\right]. \tag{31}$$

Finally, the last fundamental aspect of the model corresponds to the policy π. The policy governs the actions to be taken, given the state of the environment. It is a function that maps the state space to a probability distribution over the action space (or deterministic). Furthermore, the policy is stationary, meaning it is time-invariant, i.e., $\pi(a|s_t) \overset{d}{=} \pi(a|s_{t+h})$ if $s_t = s_{t+h}$ for all h. A deterministic policy is denoted as $\pi(s)$, where only a single action is selected for each state.

As the return of a trajectory corresponds to a realization and both the environment and the policy are stochastic, for a given policy and initial state, the trajectory represents a realization of a stochastic process.

The aim is to quantify the expected return, which is called value. To do so, we define the state value and state-action value.

The state value $V(s)$ is defined as the expected return when the agent starts in state s and acts according to policy π:

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{h=0}^{\infty} \gamma^h R_{t+h} | s_t = s\right] \tag{32}$$

For the state-action pair (s, a), we define the value $Q^{\pi}(s, a)$, which, unlike the state value function $V^{\pi}(s)$, depends on the action a:

$$Q^{\pi}(s, a) = \mathbb{E}\left[\sum_{h=0}^{\infty} \gamma^h R_{t+h} | s_t = s, a_t = a\right] \tag{33}$$

3.3.2. DRL Algorithms

Deep Q Network (DQN)

Deep Q Networks (Mnih et al. 2013[21], Mnih et al. 2015[22]) approximates the Q value function to estimate the expected reward of the agent for a given state-action. This function approximation is a neural network with parameters θ. So with the $Q-$Value approximation Q_θ, we minimize the mean square error between the current and target Q to derive the optimal state-action value function:

$$L(\theta) = \mathbb{E}\left[(Q_\theta(S, A) - Q'_\theta(S, A))^2\right] \tag{34}$$

$$Q'_\theta(S_t, A_t) = r + \gamma \arg\max_{A'} Q_\theta(S_{t+1}, A_{t+1}) \tag{35}$$

with the objective function $L(\theta)$. This is the general framework of DQN algorithms,

Policy Optimization

Proximal Policy Optimization (PPO) algorithm (Schulman et al. 2017[23]) is a policy gradient algorithm that aims to maximize the expected cumulative reward by optimizing the policy directly. Assuming the θ the parameters of the network o the policy $\pi_\theta(A, S)$. Maximising the expected cumulative reward $J(\theta)$ with a gradient ascent to learn θ:

$$J(\theta) = \mathbb{E}\left[\sum_{t=0}^{T-1} R_{t+1} | \pi_\theta\right] \tag{36}$$

$$\nabla_\theta J(\theta) = \sum_{t=0}^{T-1} \nabla_\theta \ln[\pi_\theta(A_t|S_t)] G_t \tag{37}$$

where G_t is the expected discounted cumulative rewards (30). Policy gradient methods directly learn the policy and can output a probability distribution over the actions. These methods are useful for designing stochastic policies or continuous action spaces.

Advantage Actor-Critic (A2C)

Advantage Actor-Critic (Mnih et al. 2016[24]) The A2C method is proposed to solve the training problem from policy gradient methods by updating the policy in real-time. There are two networks involved, one is the actor network that outputs the policy and the other network is the critic component that measures how good the chosen action is given the state. We can update the policy network $\pi(A|S, \theta)$ by maximising the objective function:

$$J(\theta) = \mathbb{E}\left[\ln[\pi(A|S, \theta)] A_{adv}(S, A)\right] \tag{38}$$

where $A_{adv}(S, A)$ is the advantage function defined as:

$$A_{adv}(S_t, A_t) = R_t + \gamma V(S_{t+1}|w) - V(S_t|w) \tag{39}$$

To compute the advantages of another network, the critic network with parameters w to model the state value $V(s|w)$ is used. Therefore the update of the critic network is done by gradient descent to minimize the temporal difference error:

$$J(w) = (R_t + \gamma V(S_{t+1}|w) - V(S_t|w))^2 \tag{40}$$

The A2C is useful in continuous action spaces as policy variance is constrained by using the advantage function and updates the policy in real-time.

3.4. Training Configuration

The training process is as follows: First, define de arbitrage scenario

- Step 1: define a time window of length W of historical data to estimate the cointegration portfolios.
- Step 2: use the estimated cointegration vector and the historical data to construct the arbitrage portfolios **A** and **B**. The Johansen Cointegration rank test chooses the cointegration rank at a confidence of 90%.
- Step 3: compute the environment states with the historical data and arbitrage portfolios.
- Step 4: define a trading horizon of length T to trade the arbitrage portfolios.

A historical candle window with a frequency of 30 minutes was used for scenario generation. The training period starts from 2020-11-01 00:00:00 until 2022-10-11 05:30:00, comprising 34,000 price candles for the 14 assets under study. In this way, a window of 336-time intervals is used to generate a trading scenario. This interval is divided into two components: the first 288 values are used for portfolio construction and states, and subsequently, starting from the last value, there are 48 intervals to make 47 decisions, resulting in over 30,000 distinct scenarios.

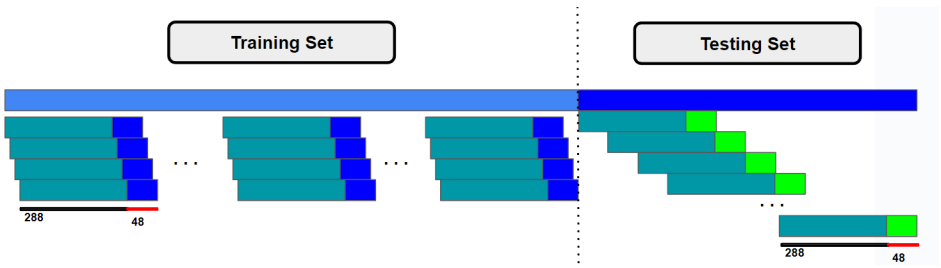

Figure 2: Training and testing scenario segmentation scheme. In the training set, there is an overlap of scenarios, while in the testing set, overlap does not occur, and no information used for training is present.

Subsequently, the testing period starts on 2022-10-19 at 07:30:00 and ends on 2023-03-18 at 07:30:00, completing a total of 150 days of transactions.

To train the DRL models, Python 3.8 programming language is used. The agents, namely Advantage Actor-Critic (A2C) (Mnih et al., 2016 [24]), Deep Q Network (DQN) (Mnih et al., 2013[21]; Mnih et al., 2015[22]), and Proximal Policy Optimization (PPO) (Schulman et al., 2017[23]), are trained on the Stable Baselines3 library.

A gamma factor of 0.995 was used, and the MlpPolicy was employed, corresponding to a fully connected dense feed-forward neural network with ReLu activation functions that return action probabilities.

Regarding training quantity, 1e6 steps were trained, with each scenario consisting of 47 steps. This means that approximately 21,000 scenarios were trained on. Additionally, the scenarios were randomly selected from the sample with equal probability.

4. Empirical Analysis

The results are presented considering the daily performance of the strategies as a result of the intra-daily decision-making process. In addition, risk and risk-adjusted return measurements are performed using different metrics.

VaR and ES risk measures consider tail risks or extreme losses incurred at a 5% level from the unconditional distribution of returns. These measures are not influenced by returns in the upper tail of the distribution; they measure extreme values at a predefined percentile, usually 5% or 1%, which serve as thresholds to determine the worst-case scenarios without considering extreme positive scenarios. These measures are estimated based on the empirical distribution.

Standard deviation is a popular and easily understandable risk measure. It represents the level of variability or scale at which returns are obtained. While it accurately measures variability, it is easily influenced by positive returns, particularly extreme positive returns, which do not necessarily indicate higher risk.

Another relevant measurement is Drawdown, which measures peak-to-peak losses, i.e., accumulated losses from the portfolio's highest value. This measurement is crucial when evaluating strategies in extreme risk environments, as it measures the percentage amplitude of the generated losses.

Finally, considering that strategies may differ in their higher moments, which can determine whether one strategy is better than another in terms of generating profitability relative to the associated risk, they are measured through the Aumann-Serrano coefficient, AS. The risk-adjusted performance is then compared by considering the four moments of the distribution through the economic performance measure, EPM.

4.1. Data

Regarding the assets used, studying their performance from 2022-01-01 to the date 2023-03-17, encompassing a total of 440 trading days with a daily granularity. A summary of statistics is presented in Table 1 for a total of 14 assets; Bitcoin (BTC), Ethereum (ETH), BNB, Ripple (XRP), Cardano (ADA), Solana (SOL), Polkadot (DOT), Bitcoin cash (BCH), Litecoin (LTC), Avalanche (AVAX), Algorand (ALGO), AAVE, UniSwap (UNI), PancakeSwap (CAKE).

The asset with the highest daily loss during the period is Solana (SOL) with a minimum return of -0.549, corresponding to a one-day loss of 42.25% in percentage terms. On the other hand, Bitcoin (BTC) has the lowest maximum loss with a one-day maximum loss of 15.38%. As for the maximum gains, Aave (AAVE) has the highest one-day maximum gain at 32.79%, while BNB has the lowest maximum gain at 13.73%.

It is observed that during the period, all assets have a negative average return, and they are all negatively skewed except for Ripple (XRP), which has a positive skewness coefficient of 0.05. Regarding the excess kurtosis values, all assets exhibit some level of excess kurtosis, but Solana stands out with an excess of 14.

The asset with the lowest correlation is Ripple (XRP) with PancakeSwap (CAKE), with a correlation of 69%. In general, the assets show a high correlation, as observed in the previous graphs. This makes it challenging to achieve diversification and reduce systemic risk when constructing portfolios solely with long positions in these

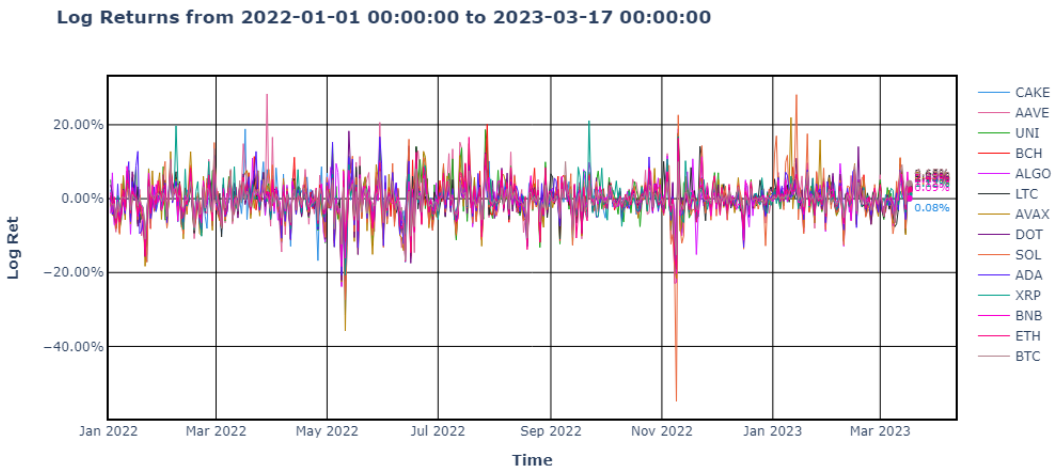

Figure 3: Log returns of the 14 assets for a period of 440 days starting on 2022-01-01 at 00:00 hours.

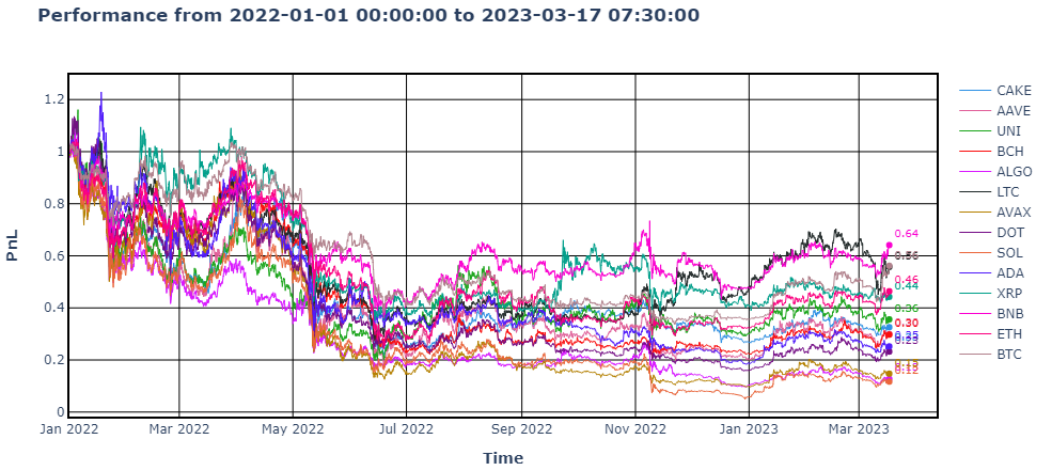

Figure 4: Performance of the 14 assets considering their values normalized to the closing price on 2022-01-01 at 00:00 hours. This corresponds to a sample of a total of 440 trading days.

assets. Additionally, considering the high degree of volatility in these assets, it creates a suitable market for arbitrage activities in short time intervals, maximizing profitability based on short-term inefficiencies.

Studying the daily returns and performances of the assets for the testing period:

In both charts, both in the testing period and in the broader market view, it can be observed that although the market is highly correlated, in the very short term, the normalized price series of all assets exhibit a coordinated movement that later begins to diverge. However, they still maintain the same overall direction of movement, considering that at specific moments, an asset deviates from the market trend and then remains at a higher or lower level. This reaffirms the potential of statistical arbitrage strategies to generate returns outside the market trend in a highly risky environment.

4.2. Risk and Performance Measurement

The Sharpe ratio is one of the most popular performance metrics in which the expected return is adjusted against another alternative, usually a risk-free rate or more generally the expected return of a low-risk alternative

	SOL	AVAX	ALGO	CAKE	AAVE	DOT	UNI	ETH	ADA	BNB	XRP	LTC	BCH	BTC
mean	-0.5	-0.4	-0.5	-0.3	-0.3	-0.4	-0.2	-0.2	-0.3	-0.1	-0.2	-0.1	-0.3	-0.1
std	6.5	5.8	5.1	4.6	6.1	4.8	5.3	4.3	4.7	3.7	4.2	4.5	4.3	3.3
min	-54.9	-35.8	-26.2	-31.5	-24.3	-22.3	-21.4	-19.1	-20.7	-20.3	-20.7	-20.9	-18.3	-16.7
50%	-0.5	-0.3	0.3	0.0	-0.1	-0.2	0.0	-0.1	-0.2	-0.1	0.0	-0.1	0.0	-0.1
max	28.2	22.0	18.9	18.9	28.4	18.3	18.7	16.6	16.7	12.9	21.1	17.7	20.1	13.5
VaR 5%	-10.1	-10.0	-8.6	-6.9	-10.6	-8.0	-9.1	-7.2	-7.9	-6.0	-6.5	-7.6	-7.5	-5.4
ES 5%	-15.7	-14.3	-13.1	-12.8	-14.5	-12.0	-12.5	-11.1	-11.7	-9.4	-10.0	-11.0	-11.3	-8.7
skew	-1.23	-0.61	-0.92	-1.25	-0.07	-0.49	-0.20	-0.38	-0.23	-0.85	0.05	-0.36	-0.35	-0.47
kurt	12.36	4.12	3.85	7.37	2.52	2.88	1.49	2.91	2.37	4.65	5.17	2.51	3.18	4.30

Table 1: Summary statistics of the sample of 440 daily logarithmic returns as percentages The Value at Risk and Expected Shortfall are derived from the empirical distribution of the obtained data.

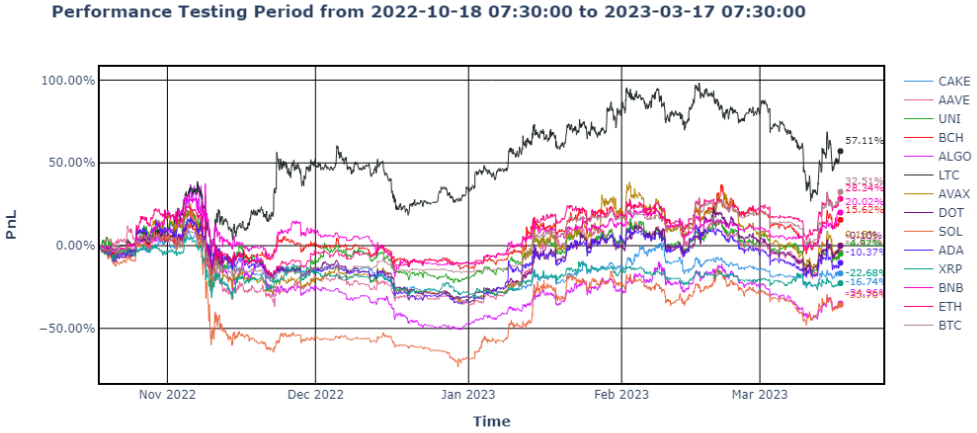

Figure 5: Performance for the selected assets in the 150 days of the testing period.

investment, over the standard deviation of the return. This metric assumes that the distribution of returns can be categorized into the family of location and scale distributions. Therefore, the fundamental elements of this metric correspond to a location parameter (the mean) and a scale parameter (standard deviation) that serves as a measure of risk. However, there are more characteristics of distribution than just its location and scale. These include higher-order moments such as skewness and kurtosis (shape parameter), which are equally important. It is known that returns exhibit characteristics in these higher moments; they are negatively skewed and have high kurtosis, greater than a normal distribution. Therefore, since the Sharpe ratio only considers the first two moments, it ignores the characteristics of the higher moments. For this reason, a metric is proposed that takes into account these elements.

The proposed metric, in contrast to the Sharpe ratio, seeks strict monotonicity with respect to stochastic dominance, so it cannot be established solely based on the location and scale of the distribution. The proposed Economic Performance Measure (EPM) metric is obtained by dividing the expected value of an investment opportunity by its Aumann & Serrano (2008) economic risk index [25], Ulrich & Pigorsch (2012) [26], denoted as the AS index. If the returns are normally distributed, then the metric coincides with the Sharpe ratio in terms of order (if $r \sim F \xrightarrow{d} N(\mu, \sigma^2)$, then EPM $\rightarrow$ 2Sharpe^2). Additionally, this metric is suitable for both high and low frequencies, while the Sharpe ratio is more appropriate for low frequencies. To this end, both parametric and non-parametric moment estimators are proposed for estimating the EPM. For parametric estimation, it is

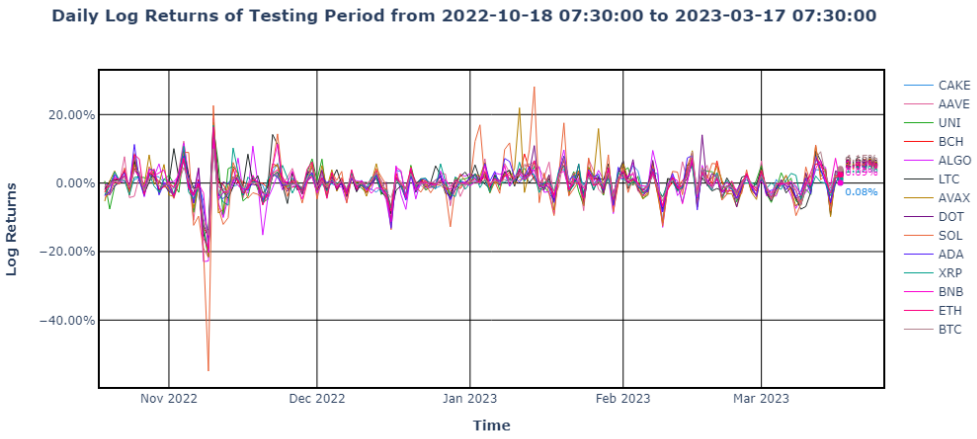

Figure 6: Log returns of the selected assets for the testing period.

proposed that the returns follow a generalized hyperbolic distribution (GHYP), which can model asymmetry and heavy tails. Thus, higher-order moments are considered for the EPM metric in a straightforward and interpretable manner. One of the results of the metric is that for returns with relatively high skewness and low kurtosis, the metric establishes a higher order compared to the Sharpe ratio.

To contrast the different implemented strategies, various performance criteria will be used. Firstly, the final profitability of the strategy (PnL) is recorded, followed by risk-reward metrics such as the Sharpe ratio, the Calmar ratio, and the Economic Performance Measure (EPM) using a generalized hyperbolic distribution.

4.2.1. Economic Performance Measure

Considering $\tilde{r} \sim F$ and r^f the risk-free rate, then $r = \tilde{r} - r^f \sim F$ corresponds to the excess return. The Economic Performance Measure (EPM) metric relative to the AS risk index is defined as follows:

$$\text{EPM}(r) = \frac{E(r)}{AS(r)} = \frac{E(\tilde{r}) - r^f}{AS(\tilde{r} - r^f)} \tag{41}$$

Considering $M(t)$ as the Moment Generating Function (MGF) of the returns, the AS index is the value $s > 0$ such that $M(-1/s) = 1$, and we denote it as $AS(r) = \{s > 0 : M(-1/s) = 1\}$. Assuming that a Generalized Hyperbolic Distribution generates r:

$$r \sim \text{GH}(\lambda, \chi, \psi, \mu, \sigma^2, \gamma), \quad r \overset{d}{=} \mu + W\gamma + \sqrt{W}A\mathbf{Z} \tag{42}$$

Where $W \sim \text{GIG}(\lambda, \chi, \psi)$ (Generalized Inverse Gaussian), $\mathbf{Z} \sim N(0, 1)$. The parameters λ, χ, ψ determine the shape of the distribution, i.e. how much weight is assigned to the tails and the center. In general, for large values of these parameters, the distribution approaches a normal distribution. Meanwhile, μ corresponds to the location parameter and σ^2 corresponds to the scale parameter. Finally, the parameter γ corresponds to the skewness parameter, where $\gamma = 0$ represents a symmetric distribution. Thus, the stochastic representation is given as follows:

$$r|_W \sim N(\mu + W\gamma, W\sigma^2) \tag{43}$$

Then we have that $E(r) = \mu + E(W)\gamma$ and $Var(r) = \gamma^2 Var(W) + E(W)\sigma^2$. Next, the probability density function is given by:

$$f_r(x) = \int_0^\infty f_{r|W}(x|w) f_W(w) dw = \int_0^\infty \frac{e^{\frac{(x-\mu)\gamma}{\sigma^2}}}{\sqrt{2\pi\sigma^2 w}} \exp\left\{-\frac{Q(x)}{2w} - \frac{w\gamma^2}{2\sigma^2}\right\} f_W(w) dw \tag{44}$$

$$f_r(x) = \frac{(\sqrt{\psi/\chi})^{\lambda}(\psi+\gamma^2/\sigma^2)^{1/2-\lambda}}{\sqrt{2\pi\sigma^2}\mathbf{K}_{\lambda}(\sqrt{\chi\psi})} \times \frac{\mathbf{K}_{\lambda-1/2}\left(\sqrt{(\chi+Q(x))(\psi+\gamma^2/\sigma^2)}\right)e^{\frac{\gamma(x-\mu)}{\sigma^2}}}{\left(\sqrt{(\chi+Q(x))(\psi+\gamma^2/\sigma^2)}\right)^{1/2-\lambda}} \tag{45}$$

where $Q(x) = (x-\mu)^2/\sigma^2$ and $\mathbf{K}_{\lambda}(\cdot)$ is a modified Bessel function of the third kind. The moment generating function (MGF) can be easily obtained since it is a mixture of normal distributions, thus:

$$\mathbf{M}_{GH}(t) = E\left[E\left[e^{tr}|W\right]\right] = e^{t\mu}E(\exp(W(t\gamma+1/2t^2\sigma^2))) \tag{46}$$

$$= e^{t\mu}\left(\frac{\psi}{\psi-2t\gamma-t^2\sigma^2}\right)^{\lambda/2}\frac{\mathbf{K}_{\lambda}(\sqrt{\psi(\chi-2t\gamma t^2\sigma^2)})}{\mathbf{K}_{\lambda}(\sqrt{\chi\psi})}, \quad \chi \geq 2t\gamma+t^2\sigma^2 \tag{47}$$

Following the EPM using a NIG, the GH has the particular case that it coincides with the NIG when $\lambda = -1/2$. Then, following Tiantian Li, Young Shin Kim, Qi Fan, and Fumin Zhu (2021)[27], defining the Aumann-Serrano coefficient $R_{AS}(X)$ as the solution to $E(\exp(-X/R)) = 1$, we have that $R_{AS}(r) = \sigma^2/(\gamma+\sqrt{\gamma^2+\sigma^2\psi})$. Therefore:

$$\text{EPM}_{GH}(r) = \frac{E(r)}{R_{AS}(r)} = \frac{E(r)(\gamma+\sqrt{\gamma^2+\sigma^2\psi})}{\sigma^2} \tag{48}$$

4.2.2. Maximal Drawdown

Considering $W(t)$ as the total wealth at time t, the drawdown is the accumulated loss from the peak wealth until the current time t, so it measures the current loss with respect to the maximum wealth achieved.

Therefore, the maximum drawdown is defined as:

$$\text{MDD} = \max_{0\geq\tau\geq T}\left\{\max_{0\geq t\geq\tau}\frac{W(t)-W(\tau)}{W(t)}\right\} \tag{49}$$

finally, we have the Calmar ratio as:

$$\textbf{Calmar :} \ \frac{\mathbb{E}[r]}{\text{MDD}} \tag{50}$$

4.2.3. Value at Risk

It corresponds to the maximum expected loss with a probability α over a period.

$$VaR_{\alpha}(X) = \inf\{x : F_X(x) \geq \alpha\} = F_X^{-1}(\alpha) \tag{51}$$

4.2.4. Expected Shortfall

The Expected Shortfall corresponds to the expected loss given that the loss exceeds the α-quantile of VaR (Value at Risk).:

$$ES_{\gamma}(X) = \frac{1}{\gamma}\int_0^{\gamma} VaR_{\alpha}(X)d\alpha \tag{52}$$

Where $ES_{\gamma}(X)$ is the Expected Shortfall of the random variable X at a level γ, which means that given a loss r, we have $ES = E[r|r < VaR]$.

4.3. Daily Performance

Generating the results for the different out-of-sample arbitrage strategies, we consider 150 trading days.

The strategies generated from the PPO, A2C, and DQN agents correspond to portfolio position directions that are changed or maintained at 30-minute intervals during a single day of trading. The COIN strategy represents the base arbitrage strategy from which improvement is expected through deep reinforcement learning. The strategy named 'Action = 0' corresponds to generating an arbitrage portfolio and maintaining a long and

short position without changes for a day. This serves as a reference considering that no actions are taken to maximize returns. It can be observed that, compared to the other strategies, the 'Action = 0' strategy, being a neutral position, maintains its purpose with minimal disturbance from market movements. This demonstrates that the generated portfolios can maintain a neutral position over time as long as they are rebalanced daily. A summary of statistics for the log-returns of the testing period is presented in Table 2, and the results of the risk and performance metrics for the strategies are presented through Table 3.

	PPO	DQN	A2C	COIN	Action = 0	BTC	LTC	SOL
mean	0.0045	0.005	0.0039	0.0048	0.0001	0.0019	0.003	-0.0029
std	0.0081	0.0076	0.0076	0.0074	0.0083	0.0277	0.0459	0.0719
min	-0.0238	-0.0156	-0.0101	-0.0106	-0.0243	-0.0851	-0.1192	-0.3712
50%	0.0036	0.0047	0.0033	0.0043	0.0002	0.0008	0.0012	-0.0045
max	0.0531	0.0404	0.05	0.0365	0.027	0.1072	0.2554	0.3327
VaR 5%	-0.0055	-0.0069	-0.0063	-0.0066	-0.0127	-0.0395	-0.065	-0.0994
ES 5%	-0.0096	-0.0104	-0.0079	-0.0079	-0.0167	-0.0579	-0.0931	-0.1632
skew	1.3872	0.5246	1.7443	0.9152	0.1308	0.6646	1.0851	-0.031
kurt	8.6045	2.6451	8.224	2.2967	0.4272	4.1571	6.4271	7.7479

Table 2: Summary of daily logarithmic returns statistics for different strategies, considering three assets of particular interest.

	MDD	ES 5%	VaR 5%	σ	AS	Calmar	Sharpe	EPM	R[%]
PPO	2.35%	-0.964%	-0.555%	0.806%	0.00444	0.192	0.561	0.816	97.20
DQN	1.55%	-1.039%	-0.695%	0.758%	0.00388	0.325	0.664	1.220	112.82
A2C	1.08%	-0.788%	-0.635%	0.760%	0.00375	0.363	0.513	0.870	79.52
COIN	1.06%	-0.787%	-0.656%	0.739%	0.00386	0.454	0.648	1.102	105.10
A = 0	3.22%	-1.674%	-1.267%	0.834%	0.00475	0.003	0.013	0.051	1.69
BTC	26.30%	-5.792%	-3.948%	2.775%	0.01558	0.007	0.068	0.048	32.51
LTC	34.82%	-9.312%	-6.504%	4.591%	0.02902	0.009	0.066	0.040	57.11
SOL	74.78%	-16.316%	-9.940%	7.189%	0.03629	-0.003	-0.040	-0.124	-35.70

Table 3: Summary of risk and performance metrics for the different agents. Additionally, the three assets BTC, LTC, and SOL are shown as market benchmarks for reference.

In terms of risk measures, during the testing period, the base strategy COIN achieves the lowest risk as measured by Maximum Drawdown, Expected Shortfall at the 5% level, and standard deviation. As for the PPO agent, it exhibits the highest risk in three metrics, while the DQN agent ranks second highest in terms of extreme losses measured by Expected Shortfall and Value at Risk. However, compared to the market, considering the best-performing asset during the period (LTC), the worst-performing asset (SOL), and a more established asset serving as the market benchmark (BTC), all strategies show a substantial improvement in terms of risk reduction. Although the strategies allocate half of the capital in long positions and the other half in short positions, while the market's "Buy & Hold" strategies have a long position with the entire capital, they are not directly comparable. Nonetheless, the strategies achieve much lower risk than the market. Considering

the neutral strategy (A = 0), similar risks are obtained compared to the active strategies, demonstrating the effectiveness of the portfolio construction strategy.

In terms of performance measures, when considering the risk-adjusted return based on different metrics, the strategy generated by the DQN agent proves to be the best during the period in terms of risk and return generated. With similar risks among the strategies, the cumulative return becomes the determining factor.

4.4. Intraday Performance

While the results are presented at a daily level, they emerge from a trajectory of 47 decisions based on different criteria. Additionally, the intraday risks and the behavior of the strategies throughout their sequence of decisions are also studied to determine the efficiency of dollar-neutral positions and whether the agents manage to generate a significant difference in terms of intraday risk compared to the benchmark strategy COIN.

The results for intraday movements of the strategies exhibit the same trend as the aggregated results for a day. Firstly, all the means are positive, and even their medians (50th percentile) are positive, unlike the neutral strategy and the comparison assets. As expected, the minimum and maximum returns are less extreme than the reference assets. On the other hand, when examining the correlations of the strategies, it is observed that all strategies, including the neutral strategy, do not show a significant correlation with the reference assets. This reinforces the fact that the portfolio construction is correct and that they have a beta very close to 1, so by taking opposing positions, a null beta is obtained. Table 4 summarizes statistics for the testing period intraday results. Table 5 presents the Pearson correlation matrix for the given strategies and benchmarks.

	PPO[%]	DQN[%]	A2C[%]	COIN[%]	A=0[%]	BTC[%]	LTC[%]	SOL[%]
mean	0.0093	0.0104	0.0081	0.0099	-0.0002	0.004	0.0064	-0.0059
std	0.125	0.1249	0.1251	0.125	0.1254	0.3829	0.6473	1.0169
min	-1.152	-1.152	-1.152	-1.152	-1.3438	-4.99	-9.8114	-13.9475
50%	0.0015	0.0019	0.0003	0.0023	0.0	-0.0017	0.0	0.0
max	2.5711	2.5711	2.5711	2.5711	2.5711	5.9289	10.6266	14.5611
VaR 5%	-0.133	-0.1293	-0.1342	-0.132	-0.1466	-0.45	-0.8404	-1.2111
ES5%	-0.2173	-0.2108	-0.2171	-0.2104	-0.2661	-0.8848	-1.4466	-2.299
skew	3.9495	4.0539	4.1584	4.1946	1.7761	0.7441	0.7746	0.2941
kurt	53.6798	53.6576	53.6325	53.6066	54.2627	35.6178	39.1638	39.5361

Table 4: Summary statistics for intraday logarithmic returns at a 30-minute frequency. Results are presented at a percentage level.

Compared to the daily results, regarding the selected risk metrics, the least risky agents are DQN and the benchmark strategy COIN, while the agents PPO and A2C turn out to be riskier. However, the results show that not only at the aggregated daily level but also at the intraday level, the risk is substantially reduced, being around one-third of the risk of Bitcoin. Additionally, the efficiency of the arbitrage portfolio is maintained when controlling for the strategy A=0.

In terms of their performance metrics, the DQN and COIN strategies continue to have the best performance, with DQN outperforming in the Sharpe ratio and COIN in Calmar and EPM. In this case, the A2C agent performs the worst in all metrics.

	PPO	DQN	A2C	COIN	A=0	BTC	LTC	SOL
PPO	1.0	0.65	0.6493	0.7208	0.2874	0.0112	0.0465	0.0065
DQN	0.65	1.0	0.7121	0.7519	0.1248	0.0035	0.0355	0.0036
A2C	0.6493	0.7121	1.0	0.7354	-0.0385	0.0127	0.0503	0.0159
COIN	0.7208	0.7519	0.7354	1.0	0.1861	-0.0009	0.0431	-0.0085
A=0	0.2874	0.1248	-0.0385	0.1861	1.0	-0.0009	0.0361	-0.0105
BTC	0.0112	0.0035	0.0127	-0.0009	-0.0009	1.0	0.6737	0.699
LTC	0.0465	0.0355	0.0503	0.0431	0.0361	0.6737	1.0	0.5902
SOL	0.0065	0.0036	0.0159	-0.0085	-0.0105	0.699	0.5902	1.0

Table 5: Correlation matrix for the intra-day log returns for the different strategies and benchmarks.

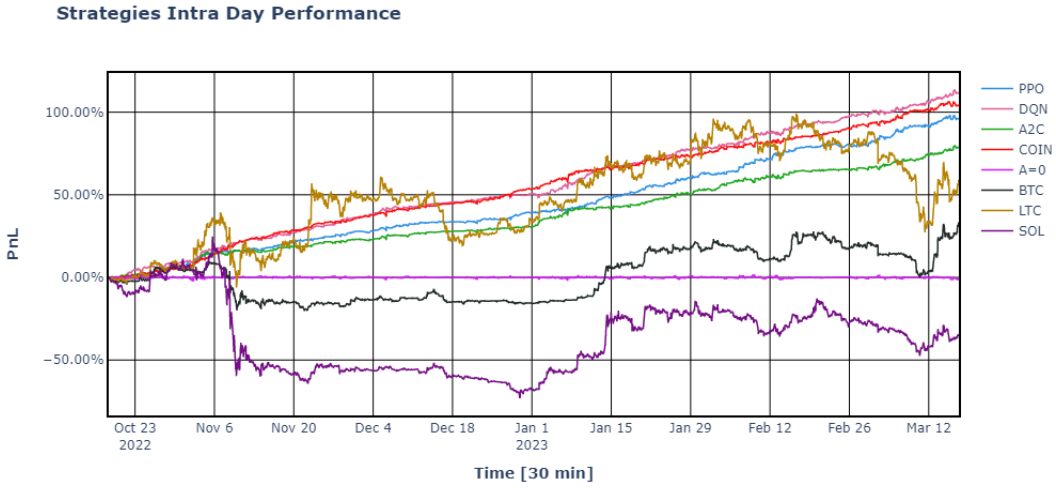

Figure 7: Profit & Loss chart of intraday strategies at a 30-minute frequency. The chart includes the neutral strategy A=0, strategies implemented by DRL agents (PPO, DQN, and A2C), and the benchmark strategy COIN.

4.5. Transaction Costs

Measuring the performance of strategies under transaction costs, considering that the strategy applies for one day, possibly restructuring the position multiple times during the day, with a maximum of 47 times in a day. Although the agents were not trained in considering transaction costs, the impact of these costs on high-frequency strategies is studied. Table 7 summary of statistics for the results when considering transaction costs. Table 8 summarizes the selected risk and performance metrics when considering transaction costs.

In comparison to the strategies without considering transaction costs, this time the agent trained by DQN outperforms the benchmark strategy COIN in all performance metrics. Regarding risk measures, the DQN agent performs the best in three risk metrics and the worst in one. Concerning tail risks or maximum losses generated, the VaR and ES metrics show that DQN exposes itself to higher losses than the other strategies. However, in terms of VaR, the differences among all strategies are minimal. Regarding unconditional volatility, all strategies have practically identical results except for PPO, which is 0.05% more volatile. As for the maximum drawdown, the DQN strategy has the lowest accumulated maximum loss in the sample, while A2C has the highest loss. This measure shows that although they all have similar levels of volatility and tail risks, there are differences in their management and ability to generate returns by controlling losses beyond a particular trade.

	MDD	ES 5%	VaR 5%	σ	AS	Calmar	Sharpe	EPM
PPO	2.526%	-0.2173%	-0.1330%	0.1250%	0.00092	0.0037	0.0747	0.01646
DQN	2.235%	-0.2108%	-0.1293%	0.1249%	0.00092	0.0047	0.0833	0.02049
A2C	2.421%	-0.2171%	-0.1342%	0.1251%	0.00091	0.0033	0.0644	0.00306
COIN	1.983%	-0.2104%	-0.1320%	0.1250%	0.00091	0.0050	0.0792	0.02547
A = 0	4.035%	-0.2661%	-0.1466%	0.1254%	0.00091	0.0000	-0.0013	0.00000
BTC	27.023%	-0.8848%	-0.4500%	0.3829%	0.00215	0.0001	0.0104	-0.00778
LTC	36.117%	-1.4466%	-0.8404%	0.6473%	0.02139	0.0002	0.0098	0.00303
SOL	78.471%	-2.2990%	-1.2111%	1.0169%	0.08455	-0.0001	-0.0058	-0.00065

Table 6: Summary of risk and performance metrics for the different agents for intraday returns. Strategies with lower risk based on the comparison metric are colored green, while those with higher risk are colored red. Additionally, the three assets BTC, LTC, and SOL are shown as market benchmarks for reference.

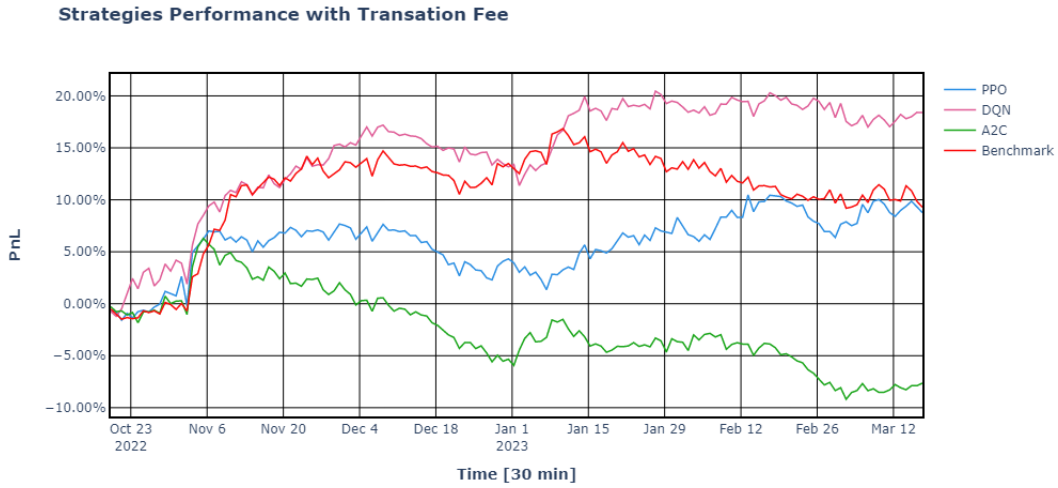

Figure 8: Profit & Loss chart for daily results of strategies when applying a transaction cost of 0.02% per trade. The Benchmark corresponds to the base COIN strategy.

In terms of the AS risk measure that considers different moments of the unconditional distribution of returns, it is found that DQN remains the least risky strategy. Considering that all strategies have similar volatility and similar extreme losses, this measure confirms what is observed in the maximum drawdown.

4.6. Agent Activity

Although the ability of the different strategies to generate returns and the level of risk they expose, along with their risk-adjusted returns, were measured, the agent's ability to predict market movements beyond whether the return is positive or negative was not assessed.

For the overall activity level, the Unconditional Coverage (UC) test by Kupiec (1995)[28] is conducted to test whether the actions are random under the null hypothesis that the actions are random with equal probability:

Under the assumption of a correct model specification, the sequence of actions should be independent and follow a Bernoulli distribution.

	PPO[%]	DQN[%]	A2C[%]	COIN[%]
mean	0.0607	0.1182	-0.0514	0.063
std	0.8217	0.7715	0.7662	0.7561
min	-2.789	-1.9423	-1.4623	-1.4953
50%	-0.0567	0.1079	-0.0891	0.016
max	4.998	3.6171	4.5487	3.2512
VaR 5%	-0.9963	-1.0925	-1.083	-1.1079
ES5%	-1.3952	-1.4484	-1.2346	-1.2236
skew	1.3834	0.4656	1.7125	0.8922
kurt	8.4942	2.2687	7.9248	2.1252

Table 7: Summary statistics for daily strategy considering transaction costs of 0.02%.

	Max D.	ES 5%	VaR 5%	σ	AS	Calmar	Sharpe	EPM	R[%]
PPO	5.87[%]	-1.39[%]	-0.99[%]	0.82[%]	0.0036	0.0103	0.0738	0.1558	8.77
DQN	4.96[%]	-1.44[%]	-1.09[%]	0.77[%]	0.0030	0.02385	0.1533	0.4004	18.39
A2C	14.57[%]	-1.23[%]	-1.08[%]	0.77[%]	0.0032	-0.0035	-0.0670	-0.1430	-7.63
COIN	6.56[%]	-1.22[%]	-1.10[%]	0.76[%]	0.0037	0.0096	0.0834	0.1597	9.29

Table 8: Summary of risk and performance metrics for the comparing strategies when transaction costs are taken.

$$H_0 : A_t \sim Bernoulli(\alpha), \tag{53}$$

$$f(A_t, p) = (1-p)^{1-A_t} p^{A_t} \tag{54}$$

For the actions $\{A_t\}_t$ where $A_t = 0$ is the action of maintaining the current position and $A_t = 1$ correspond to change direction,

$$L(\pi) = \prod_t (1-\pi)^{1-A_t} \pi^{A_t} = (1-\pi)^{T_0} \pi^{T_1} \tag{55}$$

then the maximum likelihood estimator (MLE),

$$\hat{\pi}_{MLE} = \frac{T_1}{T_0 + T_1} \tag{56}$$

T_0 is the total amount of $A_t = 0$ and T_1 to the amount of $A_t = 1$. Under the null hypothesis $H0 : \pi = \alpha$

$$LR_{uc} = -2 \ln \left[\frac{L(\alpha)}{L(\hat{\pi})} \right] \sim \chi_1^2, \tag{57}$$

Therefore, the objective is to reject the null hypothesis and conclude that the agent does not take actions randomly.

Additionally, considering the conditional distribution of actions, we have the Conditional Coverage (CC) test by Christoffersen (1998)[29].

Considering that $\{A_t\}_t$ exhibits temporal dependence and follows a Markov sequence with a transition probability matrix:

$$\Pi = \begin{pmatrix} 1-\pi_{01} & \pi_{01} \\ 1-\pi_{11} & \pi_{11} \end{pmatrix}, \tag{58}$$

where π_{ij} with $i, j \in 0, 1$ is the transition probability from state i to state j with $T_{i,j}$ the total amount of actions transitioning from state i to state j, the likelihood function is given by:

$$L(\Pi) = (1 - \pi_{01})^{T_{00}} \pi_{01}^{T_{01}} (1 - \pi_{11})^{T_{10}} \pi_{11}^{T_{11}}, \tag{59}$$

If $\{A_t\}_t$ is independent of time, then $\pi_{01} = \pi_{11} = \pi$, and we have:

$$\hat{\Pi}_{MLE} = \begin{pmatrix} 1 - \hat{\pi} & \hat{\pi} \\ 1 - \hat{\pi} & \hat{\pi} \end{pmatrix}, \quad LR_{cc} = -2 \ln \left[\frac{L(\alpha)}{L(\hat{\Pi})} \right] \sim \chi_1^2. \tag{60}$$

Therefore, we aim to reject the null hypothesis that the actions are random and independent of each other.

Agent	State 1 ($\hat{\pi}$)	State 0	$\hat{\pi}_{0,0}$	$\hat{\pi}_{0,1}$	$\hat{\pi}_{1,0}$	$\hat{\pi}_{1,1}$	UC	CC
PPO	42.44[%]	57.56[%]	55.07[%]	44.93[%]	60.92[%]	39.07[%]	<1e6	<1e6
DQN	47.79[%]	52.21[%]	50.57[%]	49.43[%]	54.02[%]	45.98[%]	<1e6	<1e6
A2C	70.27[%]	29.73[%]	31.74[%]	68.26[%]	28.89[%]	71.11[%]	<1e6	<1e6
COIN	48.31[%]	51.69[%]	48.09[%]	51.91[%]	55.52[%]	44.48[%]	<1e6	<1e6

Table 9: The Maximum Likelihood Estimation (MLE) was used to estimate the transition probabilities and unconditional probabilities relative to the number of total actions taken in each state. These actions were derived from a total of 47 trades per day for 150 days, resulting in a total of 7,050 actions executed. The results of the conducted test are reported based on their respective p-values, with the goal of rejecting the null hypothesis at a significance level of 5%.

Test results for the tests and MLE are presented in Table 9. Thus, we can reject the assumption that the actions taken by the different agents are random with equal probabilities.

5. Conclusions and further research

This paper presents a deep reinforcement learning-based trading method for generating profits in the cryptocurrency market. The method utilizes a unified framework for arbitrage portfolio generation, market signal decomposition and extraction, and decision-making to select positions on synthetic assets. The proposed method is implemented using deep reinforcement learning with three different types of agents (algorithms) for decision-making: PPO, DQN, and A2C.

Portfolios are generated based on the cointegration relationship of different assets. Market signals are extracted using various technical indicators and transformations, which define the market states for the DRL scheme. Finally, a market scenario simulator is created using a large historical window for network training. A 7-day window is used for portfolio generation, and a 1-day window (47 decisions) is used for the agent to take actions to generate profits.

An extensive out-of-sample analysis is conducted, both at the daily level considering the aggregated result of the agent's decision trajectory throughout a one-day session, and at the intra-day level considering each individual decision. The analysis focuses on the risk of the generated strategies using various risk measures and also studies the risk-adjusted performance using different metrics. The analysis is conducted both with and without transaction costs, along with a study of the actions taken by the agents to determine if there is coherent behavior rather than pure randomness.

According to the conducted tests, the proposed method is capable of generating profits and reducing risk in the highly volatile cryptocurrency market and remains profitable even when transaction costs are considered. Additionally, the results of the agent's activity show that the generated actions are not random and provide a substantial improvement (particularly with DQN) over the base arbitrage strategy (COIN).

While the results of the proposed methodology demonstrate profitability under transaction costs, the paper does not delve into the specific structure of the utilized network or consider a more granular action space.

References

[1] E. Gatev, W. N. Goetzmann, K. G. Rouwenhorst, Pairs trading: Performance of a relative-value arbitrage rule, The Review of Financial Studies 19 (3) (2006) 797–827.

[2] P. Wolf, C. Hubschneider, M. Weber, A. Bauer, J. Härtl, F. Dürr, J. M. Zöllner, Learning how to drive in a real world simulation with deep q-networks, in: 2017 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2017, pp. 244–250.

[3] H. R. Beom, H. S. Cho, A sensor-based navigation for a mobile robot using fuzzy logic and reinforcement learning, IEEE transactions on Systems, Man, and Cybernetics 25 (3) (1995) 464–477.

[4] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, et al., A general reinforcement learning algorithm that masters chess, shogi, and go through self-play, Science 362 (6419) (2018) 1140–1144.

[5] R. Neuneier, Enhancing q-learning for optimal asset allocation, Advances in neural information processing systems 10 (1997).

[6] Y. Li, Deep reinforcement learning: An overview, arXiv preprint arXiv:1701.07274 (2017).

[7] Y. Deng, F. Bao, Y. Kong, Z. Ren, Q. Dai, Deep direct reinforcement learning for financial signal representation and trading, IEEE transactions on neural networks and learning systems 28 (3) (2016) 653–664.

[8] Z. Jiang, D. Xu, J. Liang, A deep reinforcement learning framework for the financial portfolio management problem, arXiv preprint arXiv:1706.10059 (2017).

[9] J. Carapuço, R. Neves, N. Horta, Reinforcement learning applied to forex trading, Applied Soft Computing 73 (2018) 783–794.

[10] Z. Liang, H. Chen, J. Zhu, K. Jiang, Y. Li, Adversarial deep reinforcement learning in portfolio management, arXiv preprint arXiv:1808.09940 (2018).

[11] J. Lee, R. Kim, S.-W. Yi, J. Kang, Maps: Multi-agent reinforcement learning-based portfolio management system, arXiv preprint arXiv:2007.05402 (2020).

[12] X. Wu, H. Chen, J. Wang, L. Troiano, V. Loia, H. Fujita, Adaptive stock trading strategies with deep reinforcement learning methods, Information Sciences 538 (2020) 142–158.

[13] Z. Zhang, S. Zohren, S. Roberts, Deep reinforcement learning for trading, The Journal of Financial Data Science 2 (2) (2020) 25–40.

[14] F. Liu, Y. Li, B. Li, J. Li, H. Xie, Bitcoin transaction strategy construction based on deep reinforcement learning, Applied Soft Computing 113 (2021) 107952.

[15] Y.-F. Chen, S.-H. Huang, Sentiment-influenced trading system based on multimodal deep reinforcement learning, Applied Soft Computing 112 (2021) 107788.

[16] Y.-C. Lin, C.-T. Chen, C.-Y. Sang, S.-H. Huang, Multiagent-based deep reinforcement learning for risk-shifting portfolio management, Applied Soft Computing 123 (2022) 108894.

[17] J. M. Mulvey, Y. Sun, M. Wang, J. Ye, Optimizing a portfolio of mean-reverting assets with transaction costs via a feedforward neural network, Quantitative Finance 20 (8) (2020) 1239–1261.

[18] T. Kim, H. Y. Kim, Optimizing the pairs-trading strategy using deep reinforcement learning with trading and stop-loss boundaries, Complexity 2019 (2019) 1–20.

[19] J. Guijarro-Ordonez, M. Pelger, G. Zanotti, Deep learning statistical arbitrage, arXiv preprint arXiv:2106.04028 (2021).

[20] J. Durbin, S. J. Koopman, Time series analysis by state space methods, Vol. 38, OUP Oxford, 2012.

[21] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing atari with deep reinforcement learning, arXiv preprint arXiv:1312.5602 (2013).

[22] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning, nature 518 (7540) (2015) 529–533.

[23] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, arXiv preprint arXiv:1707.06347 (2017).

[24] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, K. Kavukcuoglu, Asynchronous methods for deep reinforcement learning, in: International conference on machine learning, PMLR, 2016, pp. 1928–1937.

[25] R. J. Aumann, R. Serrano, An economic index of riskiness, Journal of Political Economy 116 (5) (2008) 810–836.

[26] U. Homm, C. Pigorsch, Beyond the sharpe ratio: An application of the aumann–serrano index to performance measurement, Journal of Banking & Finance 36 (2012) 2274–2284. doi:10.1016/j.jbankfin.2012.04.005.

[27] T. Li, Y. S. Kim, Q. Fan, F. Zhu, Aumann–serrano index of risk in portfolio optimization, Mathematical Methods of Operations Research 94 (2) (2021) 197–217.

[28] P. H. Kupiec, et al., Techniques for verifying the accuracy of risk measurement models, Vol. 95, Division of Research and Statistics, Division of Monetary Affairs, Federal, 1995.

[29] P. F. Christoffersen, Evaluating interval forecasts, International economic review (1998) 841–862.