

# Improving Portfolio Optimization Results with Bandit Networks

Gustavo de Freitas Fonseca, Lucas Coelho e Silva,  
Paulo André Lima de Castro

Autonomous Computational Systems Lab - LABSCA  
Aeronautics Institute of Technology (ITA)  
São José dos Campos, São Paulo, Brazil.

Contributing authors: [gustavo.fonseca@ga.ita.br](mailto:gustavo.fonseca@ga.ita.br);  
[lucas.coelho@ga.ita.br](mailto:lucas.coelho@ga.ita.br); [pauloac@ita.br](mailto:pauloac@ita.br);

## Abstract

In Reinforcement Learning (RL), multi-armed Bandit (MAB) problems have found applications across diverse domains such as recommender systems, healthcare, and finance. Traditional MAB algorithms typically assume stationary reward distributions, which limits their effectiveness in real-world scenarios characterized by non-stationary dynamics. This paper addresses this limitation by introducing and evaluating novel Bandit algorithms designed for non-stationary environments. First, we present the *Adaptive Discounted Thompson Sampling* (ADTS) algorithm, which enhances adaptability through relaxed discounting and sliding window mechanisms to better respond to changes in reward distributions. We then extend this approach to the Portfolio Optimization problem by introducing the *Combinatorial Adaptive Discounted Thompson Sampling* (CADTS) algorithm, which addresses computational challenges within Combinatorial Bandits and improves dynamic asset allocation. Additionally, we propose a novel architecture called Bandit Networks, which integrates the outputs of ADTS and CADTS, thereby mitigating computational limitations in stock selection. Through extensive experiments using real financial market data, we demonstrate the potential of these algorithms and architectures in adapting to dynamic environments and optimizing decision-making processes. For instance, the proposed bandit network instances present superior performance when compared to classic portfolio optimization approaches, such as capital asset pricing model, equal weights, risk parity, and Markovitz, with the best network presenting an out-of-sample Sharpe Ratio 20% higher than the best performing classical model.

**Keywords:** multi-armed bandits, portfolio optimization, non-stationary bandits, bandit networks

## 1 Introduction

In the field of Reinforcement Learning (RL), there has been a growing research interest in Multi-Armed Bandit (MAB) problems, a particular problem of RL interpreted as a tabular solution method, where storing transitions does not matter ([Charpentier et al., 2023](#)). Despite their simplicity, these problems have gained attention for their effectiveness in addressing real-world challenges, finding applications ranging from recommender systems ([Silva et al., 2022](#)), human search behavior ([Nakazato et al., 2024](#)) and information retrieval ([Losada et al., 2017](#)) to domains like healthcare ([Zhou et al., 2023](#)) and finance ([Bouneffouf et al., 2020](#)).

Most of the classical MAB framework assumes stationary reward distributions, where the underlying probabilities remain constant over time. However, real-world applications often feature inherently non-stationary environments that may undergo shifts in their probability distributions. In this sense, the need to address non-stationarity arises. One real environment with such behavior is the finance field ([de Castro and Annoni, 2016](#)), where changes in market dynamics demand rapid model responses as to avoid unnecessary risk ([SBRANA, 2023](#) and [de Castro and Parsons, 2014](#)). Similarly, in online advertising, user preferences and behavior may change over time, necessitating adaptive strategies to optimize ad placement and maximize click-through rates. If one intends to use MAB to solve these problems, the non-stationary variants might be a great fit.

In such dynamic contexts, traditional algorithms falter, leading to the need for the development of novel strategies capable of adapting to changing reward structures in real-time. While current literature discusses algorithms that deal with non-stationarity in Bandits problems, existing solutions often encounter limitations regarding the temporal dynamics of policy adaptation, as current formulations exhibit varying degrees of responsiveness to environmental shifts. In this sense, there are opportunities for exploring novel modeling approaches and algorithms, especially toward improved dynamic adaptability of Bandit algorithms under non-stationary conditions.

Particular to finance, the Portfolio Optimization problem emerges as a pertinent application area for MAB solutions. Portfolio Optimization involves selecting and allocating assets to achieve a desirable balance of risk and return. Traditional approaches often rely on static allocation strategies, which may fail to account for changing market conditions effectively. By improving MAB techniques, Portfolio Optimization can benefit from adaptive allocation strategies that dynamically adjust asset weights in response to evolving market dynamics ([Chen et al., 2024](#)). However, existing literature addressing Portfolio Optimization with MABs remains sparse, highlighting a significant research gap in this domain.

The contributions of this paper are manifold. First, it addresses the challenge of non-stationarity in MAB problems by proposing the Adaptive Discounted Thompson

Sampling (ADTS) algorithm. The ADTS algorithm enhances adaptability through relaxed discounting and sliding window mechanisms, allowing it to respond to changes in reward distributions. The algorithm is evaluated through stock picking and portfolio optimization experiments, using historical data from the S&P 500 index.

Then, building on the ADTS algorithm, this work introduces the Combinatorial Adaptive Discounted Thompson Sampling (CADTS) algorithm for portfolio optimization within the framework of Combinatorial Bandits. The CADTS algorithm addresses the computational challenges associated with combinatorial bandits and aims to optimize decision-making processes in dynamic environments.

Finally, to further enhance the applicability of these algorithms, a novel architecture called Non-Stationary Bandit Networks is proposed. This architecture integrates the outputs of ADTS and CADTS, mitigating biases and improving the robustness of the stock selection process. The effectiveness of these algorithms and architectures is demonstrated through empirical evidence, showcasing their potential in optimizing financial decision-making in non-stationary environments.

In the following sections, we provide a detailed description of our research on non-stationary bandits and the network concepts. The Literature Review in Section 2 covers recent work about non-stationary bandits and the practical usage of MABs in Portfolio Optimization. In Section 3, we present a detailed description of the novel non-stationary bandit algorithm, Adaptive Discounted Thompson Sampling (ADTS), its combinatorial bandit variant, Combinatorial Adaptive Discounted Thompson Sampling (CADTS) and how they together constitute original architectures called Bandit Networks aimed to solve the Portfolio Optimization problem using historical daily price of the Standard and Poor's (S&P) stocks. Section 4 presents the experimental setup, aiming to evaluate the ADTS and the bandit network instances applied to financial market data of a set of Standard and Poor's stocks. Section 5 presents our findings through the experiments, including the stock picking, portfolio optimization, and portfolio optimization robustness experiments. In Section 6, we present a comprehensive discussion of the research and introduce practical implications of the ADTS and the bandit networks on finance. Finally, we conclude our study by summarizing the contributions and implications for future research.

## 2 Literature Review

The goal of this research is to improve the practical usage of Multi-Armed Bandits (MAB) in changing environments such as finance. This section explores the forefront advancements of Non-Stationary Bandit algorithms and outlines some practical applications of bandit algorithms in finance.

### 2.1 Non-Stationary Bandits

Unlike traditional stationary bandit settings, where rewards associated with each action remain constant throughout the learning process, non-stationary bandit problems arise in scenarios where the underlying environment is subject to stochastic and agent-independent changes over time, leading to variations in the reward distribution. ([Allesiardo et al., 2017](#)).

The need to address non-stationarity arises in real-world applications where the environment is inherently dynamic and may undergo unpredictable shifts. In finance, stock prices and returns are constantly changing, demanding fast changes to avoid unnecessary risks. Similarly, in online advertising, user preferences and behavior may change over time, necessitating adaptive strategies to optimize ad placement and maximize click-through rates. [Lattimore and Szepesvári \(2020\)](#) argues that the process of building a non-stationary bandit variant typically occurs by applying discounts or sliding windows to pre-existing stationary policies. These artifices dynamically augment the exploration components and prevent the algorithm from being locked into a local minimum.

[Raj and Kalyani \(2017\)](#) have introduced the Discounted Thompson Sampling (D TS) philosophy, to continuously increase the variance of the prior distribution and maintain exploration over time, mitigating the effect of past observations. The work also introduced the optimistic version of D TS, the so-called Discounted Optimistic Thompson Sampling (DOTS). In DOTS, the samples are forced to have at least its expected value, thus increasing the arms' exploitative value. The dTS and dOTS were challenged only in synthetic data, although showcased a good margin of regret in slow and fast varying environments, compared to other algorithms, such as the Classical Thompson Sampling [14].

[Trovò et al. \(2020\)](#) presented the Sliding-Window Thompson Sampling for non-stationary MAB settings. As the name suggests, it adapts the sampler to a hot trace by inspecting past successes and failures given a sliding window hyper-parameter. This work provides regret upper bounds for dynamic pseudo-regret in different scenarios. Empirical evidence showed that sw-TS outperforms existing algorithms in non-stationary settings.

Following this path, [Cavenaghi et al. \(2021\)](#) introduces a new Thompson Sampling variant called f-Discounted-Sliding-Window Thompson Sampling (f-dsw TS) to address concept drift problems. In this case, the work combines both the concepts of discounts and sliding windows. The discount factor adjusts the choices of a historical sampler while the sliding window walks through a short-term sampler, that is processed in parallel. These two samplers are instantiated by each arm using an aggregation function  $f(\cdot)$ . The aggregation function can compare both historical and short-term samplers based on three types of approaches: i) pessimistic, the minimum between each sampler ( $f = \min$ ), ii) optimistic, the maximum between each sampler ( $f = \max$ ) and iii) the average of the two samplers ( $f = \text{mean}$ ). The work conducts experiments in synthetic and real-world environments and compares f-dsw TS with stationary and non-stationary TS baselines. Based on the simulations, the f-dsw TS algorithm outperforms baselines in synthetic environments. The pessimistic version ( $f = \min$ ) is most effective in real-world data.

From the frequentist perspective, [Garivier and Moulines \(2011\)](#) explores the Discounted Upper Confidence Bound (D UCB) and Sliding Window Upper Confidence Bound (SW-UCB) variants of UCB. The work establishes upper and lower bounds for regret in changing environments and points out that these policies adapt well to non-stationary environments. [Cao et al. \(2019\)](#) presents an innovative M-UCB

algorithm with near-optimal regret bounds, integrating change detection with traditional UCB methods. Experimental comparisons confirm its superior performance. Liu et al. (2017) introduces CD-UCB policies, including CUSUM-UCB and PHT-UCB, showcasing regret reduction across synthetic and real datasets.

While the covered state-of-the-art Bandit policies provide attempts at tackling the concept drift in non-stationary settings, there remain unsolved problems regarding the proposition of novel non-stationary Bandit policies, especially toward improved dynamic adaptability of Bandit algorithms in such conditions.

## 2.2 Applications of Bandits in Portfolio Optimization

To fulfill our objective of providing a comprehensive overview of the applications of bandits in finance, we explore some applications in the field of finance, ranging from portfolio optimization to high-frequency trading strategies.

Portfolio Optimization is a crucial problem in finance, aiming to allocate assets to achieve optimal returns while managing risks effectively. The application of Reinforcement Learning (RL) techniques, including Bandit algorithms, has garnered significant attention in recent years (Wang, 2019). Some researchers argue that simpler online learning algorithms like bandits can effectively address the allocation problem (Li and Hoi, 2014).

Stochastic multi-armed Bandit models, which address the exploration-exploitation trade-off, offer a natural framework for sequential decision-making under uncertainty, making them suitable for portfolio selection. By incorporating risk awareness and employing optimal policies, Huo and Fu (2017) have aimed to strike a balance between risk and return in portfolio construction. Similarly, bandit algorithms have been utilized to exploit correlations among assets, leading to the development of effective online portfolio selection strategies (Shen et al., 2015).

Classic portfolio optimization models, such as Markowitz's mean-variance optimization (Markowitz, 1952), face challenges in parameter estimation and applicability across different market conditions. In contrast, bandit-based strategies offer flexibility and adaptability, particularly in environments where traditional models may falter. By treating different portfolio strategies as strategic arms in a multi-armed bandit setup, Zhu et al. (2019) has sought to maximize rewards through a judicious balance of exploration and exploitation.

In summary, the literature on real applications of bandits in portfolio optimization problems is still incipient, which highlights opportunities to contribute to the research by unifying these two different domains.

## 3 Proposal

In this Section, we present a detailed description of the novel non-stationary bandit algorithm, Adaptive Discounted Thompson Sampling (ADTS), its combinatorial bandit variant, Combinatorial Adaptive Discounted Thompson Sampling (CADTS) and how they together constitute original architectures called Bandit Networks aimed to solve the Portfolio Optimization problem using historical daily price of the Standard and Poor's (S&P) stocks. We empirically evaluate the performance of the proposed

algorithms and network architectures based on the experiments presented in the subsequent sections.

### 3.1 Adaptive Discounted Thompson Sampling

We introduce the Adaptive Discounted Thompson Sampling (ADTS), a Thompson Sampling (TS) ([Thompson, 1933](#)) variant aimed to deal with non-stationary environments more efficiently.

The TS algorithm tracks the rewards history  $X_t^k$  for each arm  $k$  using a Bernoulli distribution, denoted as  $\mathcal{B}(\alpha_k, \beta_k)$ , which has the parameters  $\alpha$  and  $\beta$ . Physically,  $\alpha$  can be interpreted as the cumulative success counts while  $\beta$  works as the cumulative failure counts. In that sense, the distribution  $\mathcal{B}(\alpha_k, \beta_k)$  yields the expected success value by pulling each arm  $k$ . The classical TS updating is governed by the expression:

$$\mathcal{B}(\alpha_k, \beta_k) = \begin{cases} \mathcal{B}(\alpha_k, \beta_k), & \text{if } I_t \neq k \\ \mathcal{B}(\alpha_k + X_t^k, \beta_k + 1 - X_t^k), & \text{if } I_t = k \end{cases} \quad (1)$$

where  $I_t$  is the selected arm at step  $t$ .

In the language of bandits, the regret  $R(t)$  represents the cumulative learning error. It quantifies the difference between an always optimal choice, or oracle ([Besbes et al., 2014](#)), and the sub-optimal choices by some bandit policy:

$$R(t) = \sum_{t=1}^T X_t^* - \mathbb{E} \left[ \sum_{t=1}^T X_t^k \right] \quad (2)$$

where  $X_t^*$  is the optimal reward at step  $t$ .

The regret  $R(t)$  measure can be used to compare and contrast different bandit policies. In non-stationary environments such as the financial stock markets, where the stock returns distributions are changing, classical bandit algorithms tend to show higher regret values, as they usually get stuck into some local optimum arm. To minimize this problem, the non-stationary bandit variants appeared.

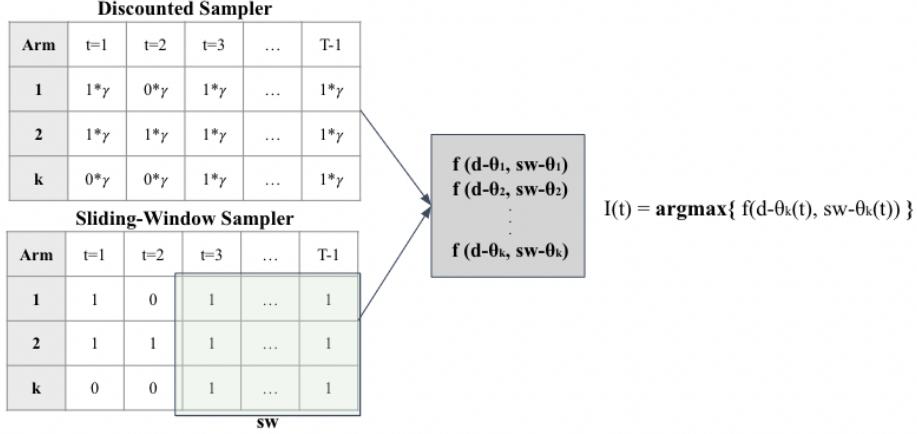
The ADTS algorithm ([de Freitas Fonseca et al., 2024](#)), adapted from ([Cavenaghi et al., 2021](#)), relaxes the application of the discount factor by applying it only for the selected arm  $I_t$ , instead of applying it for all of the arms. On the other hand, we keep intact the construction of what we interpret as the short-term memory of the policy, by applying the sliding window approach, and then comparing both discounted and short-term samples with the aggregation function  $f(\cdot)$  for each arm.

More formally, the discount factor  $\gamma \in (0, 1]$  gradually diminishes the impact of past observations in the historic trace. The short-term trace, represented as  $\check{\mathcal{B}}(\alpha_k^n, \beta_k^n)$ , tracks the recent rewards by applying a sliding window  $w$ . The mix between the historical and hot traces components is performed before the arm is played at step  $t$ . For each arm  $k$ , the algorithm computes an aggregated score  $S_k(t)$ , as:

$$S_k(t) = f(\theta_k(t), \check{\theta}_k(t)) \quad (3)$$

where  $f(\cdot)$  is the aggregation function defined for the algorithm (*min*, *avg*, *max*),  $\theta_k(t)$  is a sample from the historic trace distribution  $\mathcal{B}(\alpha_k, \beta_k)$ ,  $\check{\theta}_k(t)$  is a sample from the short-term trace distribution  $\check{\mathcal{B}}(\alpha_k^w, \beta_k^w)$  at step  $t$  for arm  $k$ . Finally, ADTS chooses

which arm to play at step  $t$  as  $I_t = \text{argmax}(S_k(t))$ . Figure 1 depicts the arms-pulling process when computing in parallel the short-term and historical traces.



**Fig. 1:** ADTS Selection Diagram.

Algorithm 1 outlines the ADTS strategy. In lines (2–5), for each arm, we sample a reward estimate from both historic and short-term distributions. In line (6) we apply the aggregation function  $f(\cdot)$  to select one of the two estimates (or a mix of them) and choose the arm with the highest aggregated score. We pull the arm and observe the reward at time  $t$  (i.e.,  $r_t$ ) in line (7). In line 9, we apply the discount factor only for the selected arm  $k = I_t$ .

---

**Algorithm 1** Adaptive Discounted Thompson Sampling TS.

---

**Input:**

$k = |\mathcal{K}| \geq 2$  number of arms

$\gamma \in (0, 1]$  Discount factor

$w \in [1, T]$  Sliding-window size

```

1: for  $t = 1, 2, \dots, T$  do
2:   for  $k = 1, 2, \dots, K$  do
3:      $\theta_k(t) = \mathcal{B}(\alpha_k + 1, \beta_k + 1)$ 
4:      $\check{\theta}_k(t) = \check{\mathcal{B}}(\alpha_k^w, \beta_k^w)$ 
5:   end for
6:   Play arm  $I(t) = \arg \max_k (f(\theta_k(t), \check{\theta}_k(t)))$ 
7:   Observe reward  $r_t$ 
8:    $X_t = 1$  if  $r_t = r_t^*$  else 0
9:   Update  $\mathcal{B}(\alpha_k, \beta_k)$  as:
10:    
$$\mathcal{B}(\alpha_k, \beta_k) = \begin{cases} (\alpha_k, \beta_k), & \forall k \neq I_t \\ \gamma(\alpha_{I_t}, \beta_{I_t}) + (X_t, 1 - X_t) & \end{cases}$$

11:   Update  $\check{\mathcal{B}}(\alpha_k^w, \beta_k^w)$ , where  $k = I_t$ , with the last  $w$  rewards taken for arm  $k$ 
12: end for

```

---

### 3.2 Combinatorial Adaptive Discounted Thompson Sampling

In this section, we extend the ADTS algorithm to a combinatorial bandit problem, originating the Combinatorial Adaptive Discounted Thompson Sampling (CADTS).

For conducting the Portfolio Optimization problem in the context of Bandits, we combined ADTS with the Combinatorial Bandits formulation proposed by [Chen et al. \(2013\)](#). Theoretically, each stock can have infinite possible weight values  $w_k$ , which can lead our CADTS to dimensionality issues. To avoid that, we construct our feasible portfolio weights combinations (superarms) by building an array of discrete weights  $pw_k$  for each stock  $k$ , given the total number of stocks  $K$ .

$$pw[k, :] = [0 \ 1s \ 2s \ 3s \ \dots \ 1] \quad (4)$$

where  $s = \frac{1}{2^K}$  is the minimum weight step value.

Then, we construct the possible weights matrix  $PW$ , where the rows represent each stock and the columns are the possible weights for each stock defined in equation 4. To create the super-arms, we run all the possible weight combinations between the stocks and possible weights in the  $PW$  matrix s.t.  $\sum_{k=1}^K w_k(t) = 1$ :

$$PW = \begin{bmatrix} \text{Stock}_1 & 0 & 1s & 2s & \cdots & 1 \\ \text{Stock}_2 & 0 & 1s & 2s & \cdots & 1 \\ \vdots & \vdots & \vdots & \ddots & & \vdots \\ \text{Stock}_k & 0 & 1s & 2s & \cdots & 1 \end{bmatrix} \quad (5)$$

Algorithm 2 outlines the complete pseudo-code for the Combinatorial Adaptive Discounted Thompson Sampling TS (CADTS). Lines (1-5) describe the procedure to generate the feasible superarms  $\mathcal{S}$  containing the combinations of weights for each stock. From lines (7-19) we repeat the non-stationary bandit problem outlined in Algorithm 1.

---

**Algorithm 2** *Combinatorial Adaptive Discounted Thompson Sampling TS.*


---

**Input:**

$K$  Number of arms  
 $s = \frac{1}{2K}$  Minimum weight step value  
 $\gamma \in (0, 1]$  Discount factor  
 $w \in [1, T]$  Sliding-window size  
 $n$  Number of arms inside each superarm  $S$

- 1: **Generate the portfolio feasible super arms ( $\mathcal{S}$  given  $K$  and  $s$ )**
  - 2: **for**  $k = 1, 2, \dots, K$  **do**
  - 3:      $pw[k, :] = [0 \ k s \ 2ks \ 3ks \ \dots \ 1]$
  - 4: **end for**
  - 5:  $\mathcal{S} = \left[ \text{Combinations}(w_{k,i}) \mid \sum_{k=1}^K pw_{k,i} = 1 \right]$
  - 6:
  - 7: **Run ADTS**
  - 8: **for**  $t = 1, 2, \dots, T$  **do**
  - 9:     **for**  $S = 1, 2, \dots, \mathcal{S}$  **do**
  - 10:          $\theta_S(t) = \mathcal{B}(\alpha_S + 1, \beta_S + 1)$
  - 11:          $\check{\theta}_S(t) = \check{\mathcal{B}}(\alpha_S^w, \beta_S^w)$
  - 12:     **end for**
  - 13:     Play arm  $I_t = \arg \max_S (f(\theta_S(t), \check{\theta}_S(t)))$
  - 14:     Observe the portfolio reward  $r_t = \sum_{k=1}^K w_k(t) r_k(t)$
  - 15:      $X_t = 1$  if  $r_t = r_t^*$  else 0
  - 16:     Update  $\mathcal{B}(\alpha_S, \beta_S)$  as:
  - 17:         
$$\mathcal{B}(\alpha_S, \beta_S) = \begin{cases} (\alpha_S, \beta_S), & \forall k \neq I(t) \\ \gamma(\alpha_{I_t}, \beta_{I_t}) + (X_t, 1 - X_t) & \end{cases}$$
  - 18:     Update  $\check{\mathcal{B}}(\alpha_S^w, \beta_S^w)$ , where  $S = I_t$ , with the last  $w$  rewards taken for super arm  $S$
  - 19: **end for**
-

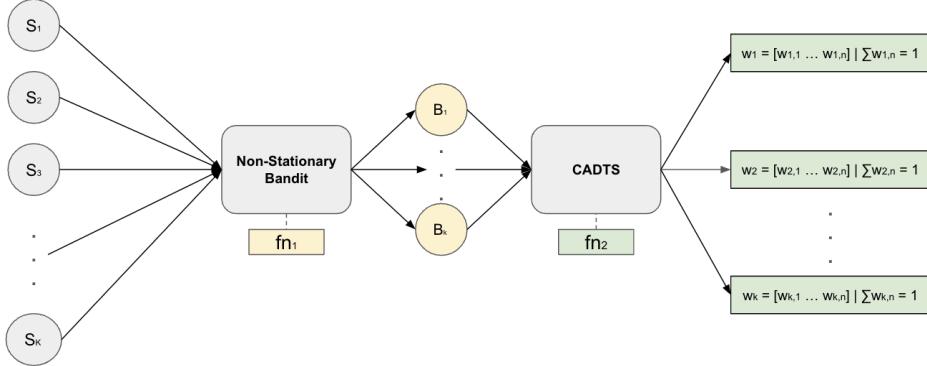
### 3.3 Bandit Networks

We demonstrated in (de Freitas Fonseca et al., 2024) that the ADTS algorithm efficiently selects the best arm in a changing environment. Heavily inspired by the Neural Networks philosophy, we introduce a novel approach called Bandit Networks. It connects between layers of non-stationary bandits policies such as ADTS and CADTS. In this section we propose two different architectures to solve the Portfolio Optimization problem: i) Non-Stationary Bandit with CADTS Network and ii) Two-layer ADTS Network.

#### 3.3.1 Non-Stationary Bandit with CADTS Network

Figure 2 displays the Non-Stationary Bandit with CADTS Network architecture. In the first layer, the non-stationary Bandit policy (ADTS, D TS, SW UCB, or any other) receives  $S_1, S_2, \dots, S_K$ , the complete universe of stocks. More than selecting the best stock at time step  $t$ , the role of the non-stationary Bandit policy is to provide the second layer the rank of the  $k < K$  best stocks, based on the reward function  $fn_1$ , colored in yellow in the diagram. The function  $fn$  can be constructed to select stocks based on historical or sliding-window cumulative returns, momentum, or risk-adjusted returns, such as the Sharpe Index.

Having the  $k$  best stocks, the CADTS generates the portfolio feasible weights combinations  $s.t. \sum_{k=1}^K pw_{k,i} = 1$ , as described in Algorithm 2. The policy is accountable for selecting at time step  $t$  the best weight combination (super arm) that maximizes its reward function  $fn_2$ , colored in green. Similarly, the second layer objective function can also be constructed to select stocks based on a financial metric, whether the same as  $fn_1$  or a different one.



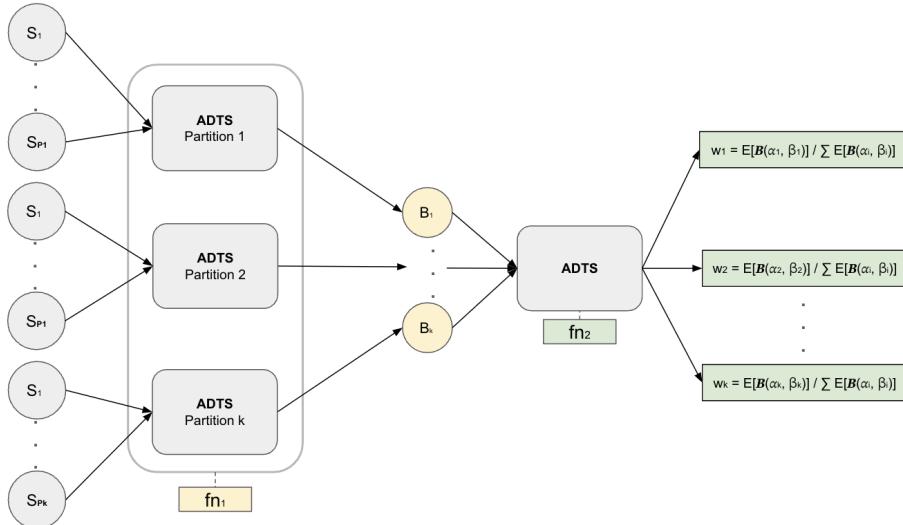
**Fig. 2:** Non-Stationary Bandit with CADTS Network architecture

### 3.3.2 Two-layer ADTS Network

We present an alternate to the Non-Stationary Bandit with CADTS Network architecture. Figure 3 displays the Two-layer ADTS Network. In the first layer, we partition the total stocks universe into  $k$  parts. For each partition, we run an ADTS to filter the stocks universe using the reward function  $fn_1$ , colored in yellow in the diagram.

Given the  $k$  best stocks to the second layer, we bolt another ADTS to learn the  $k$  best stocks hierarchy given another reward function  $fn_2$ , colored in green. The portfolio weights are generated by normalizing the expected success values of each  $k$  Bernoulli distribution:

$$w_i = \frac{\mathbb{E}[\mathcal{B}(\alpha_i, \beta_i)]}{\sum_{i=1}^k \mathbb{E}[\mathcal{B}(\alpha_i, \beta_i)]} \quad (6)$$



**Fig. 3:** Two-layer ADTS Network architecture

## 4 Experimental Setup

The experiments designed in this paper aim to demonstrate the practical usefulness of the proposed ADTS and CADTS algorithms and their connection to form the Bandit Networks in a real environment provided by a set of daily returns of S&P stocks.

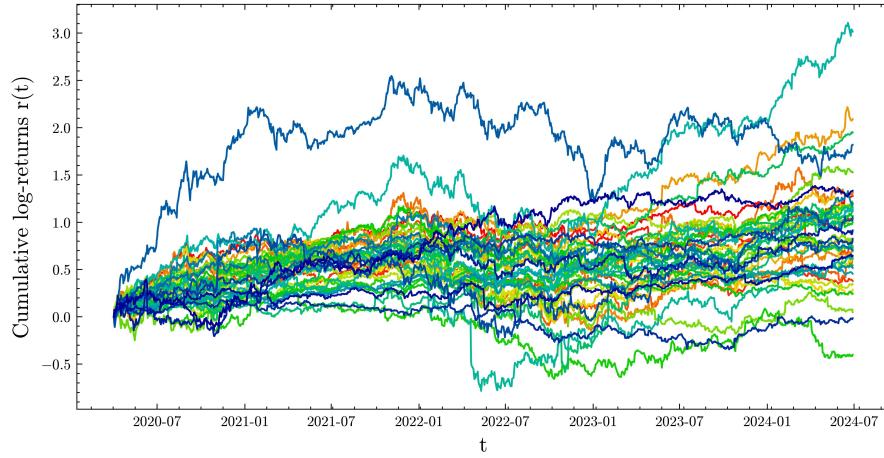
To achieve this goal, we divide the experiments into three phases. In the first phase, we investigate if the ADTS effectively selects the best S&P stock in the so-called Stock Picking problem. Next, we evaluate the performance of different Bandit Networks instances in the Portfolio Optimization Problem given a set of S&P stocks. Finally, we check the Bandit Networks instance's results robustness by removing a set of high-performing stocks.

In this section, we present the selected set of S&P stocks and the collected daily market data, describe each of the three experiments, and present the benchmarks used to compare the results.

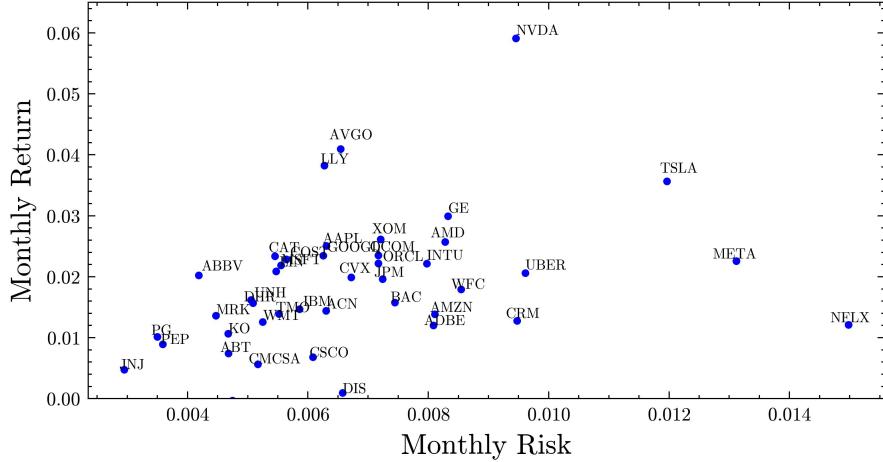
#### 4.1 Market Data and Problem Definition

We submit the non-stationary bandit's policies to a real-world problem by selecting a set of 44 stocks within the S&P index, taking their historical prices from April 2020 to July 2024. These stocks behave as our arms in a bandit problem context. Given the historical series of daily returns  $[r_0, r_1, r_2, \dots, r_t]$ , where  $t$  represent each time step, the objective is to maximize the future rewards  $[X_{t+1}, X_{t+2}, X_{t+3}, \dots, X_T]$  either for a unique stock or a portfolio of stocks. The bandit reward function is defined by  $F([r_{w_f-t}, \dots, r_{w_f-2}, r_{w_f}, r_{w_f}])$ , where  $F$  is a financial metric such as cumulative Returns, Sharpe Index, Sortino Ratio, etc and  $w_f$  represents the window length in case of applying sliding window to the financial function. If the historical financial function is desired at each time step  $t$ , the sliding window becomes infinite and no hyperparameter  $w_f$  is necessary.

The arms' logarithmic cumulative daily returns are shown in Figure 4 and the monthly log returns and risks are displayed in Figure 5. Table 1 summarizes the monthly returns and risks.



**Fig. 4:** Selected S&P stocks logarithmic cumulative daily returns



**Fig. 5:** Monthly log-return and risk for each selected stock

**Table 1:** Stocks Monthly Risks and Returns

Stock Symbol	Monthly Return $\pm$ Monthly Risk	Stock Symbol	Monthly Return $\pm$ Monthly Risk
NVDA	0.059 $\pm$ 0.009	UNH	0.016 $\pm$ 0.005
AVGO	0.041 $\pm$ 0.007	BAC	0.016 $\pm$ 0.007
LLY	0.038 $\pm$ 0.006	DHR	0.016 $\pm$ 0.005
TSLA	0.036 $\pm$ 0.012	IBM	0.015 $\pm$ 0.006
GE	0.030 $\pm$ 0.008	ACN	0.014 $\pm$ 0.006
XOM	0.026 $\pm$ 0.007	TMO	0.014 $\pm$ 0.006
AMD	0.026 $\pm$ 0.008	AMZN	0.014 $\pm$ 0.008
AAPL	0.025 $\pm$ 0.006	MRK	0.014 $\pm$ 0.004
QCOM	0.024 $\pm$ 0.007	CRM	0.013 $\pm$ 0.009
GOOGL	0.023 $\pm$ 0.006	WMT	0.013 $\pm$ 0.005
CAT	0.023 $\pm$ 0.005	NFLX	0.012 $\pm$ 0.015
COST	0.023 $\pm$ 0.006	ADBE	0.012 $\pm$ 0.008
META	0.023 $\pm$ 0.013	KO	0.011 $\pm$ 0.005
ORCL	0.022 $\pm$ 0.007	PG	0.010 $\pm$ 0.003
INTU	0.022 $\pm$ 0.008	PEP	0.009 $\pm$ 0.004
MSFT	0.022 $\pm$ 0.006	ABT	0.007 $\pm$ 0.005
LIN	0.021 $\pm$ 0.005	CSCO	0.007 $\pm$ 0.006
UBER	0.021 $\pm$ 0.010	CMCSA	0.006 $\pm$ 0.005
ABBV	0.020 $\pm$ 0.004	JNJ	0.005 $\pm$ 0.003
CVX	0.020 $\pm$ 0.007	DIS	0.001 $\pm$ 0.007
JPM	0.020 $\pm$ 0.007	VZ	-0.000 $\pm$ 0.005
WFC	0.018 $\pm$ 0.009	INTC	-0.008 $\pm$ 0.007

## 4.2 Stock Picking Experiment

In this experiment, we aim to apply the proposed ADTS algorithm to the stock picking problem, given the set of the S&P stocks from 4.1. To provide benchmark comparisons to our non-stationary bandit variant, we invoke the bandit algorithms listed below as Table 2 summarizes the complete experiment setup.

- Classical Thompson Sampling: Classical TS ([Thompson, 1933](#));
- f-Delayed-Sliding-Window Thompson Sampling: F-DSW TS ([Cavenaghi et al., 2021](#));
- Discounted Thompson Sampling: D-TS ([Raj and Kalyani, 2017](#));
- UCB-1 ([Auer, 2002](#)).
- Discounted UCB: D-UCB ([Garivier and Moulines, 2011](#));
- Sliding-Window UCB: SW-UCB ([Garivier and Moulines, 2011](#));

**Table 2:** S&P Stock Picking - Experiment Setup

Algorithm	Bandit Family	Reward Function	Window Length ( $w_f$ )	Hyper-Parameters
Classical TS	-	Mean Return	100 days	-
ADTS	TS	Mean Return	100 days	$\gamma = 0.9; f = \text{mean}; w = 100$
F-DSW TS	TS	Mean Return	100 days	$\gamma = 0.99; f = \text{mean}; w = 100$
D TS	TS	Mean Return	100 days	$\gamma = 0.99$
UCB-1	-	Mean Return	100 days	-
D UCB	UCB-1	Mean Return	100 days	$\gamma = 0.1$
SW UCB	UCB-1	Mean Return	100 days	$w = 50$

Next, we investigate the financial metrics of the experimented policies (Total Return, Sharpe Ratio, Drawdown, Win Rate and Sortino Ratio), comparing with each other and with the S&P Index.

Finally, we finish the experiment investigating the drift effect on each bandit policy by applying an artificial shock to the best performing ticker, the NVDA. We are interested to investigate how does a shock impact on each policy with respect to their financial metrics distributions.

## 4.3 Portfolio Optimization Experiment

For conducting the Portfolio Optimization for the S&P stocks presented in Section 4.1, we apply different Bandit Networks instances. Table 3 summarizes the applied Bandit Networks in the experiments, describing their architectures, algorithms, and rewards functions for each layer and the portfolio size  $k$ .

In the experiments, we analyze the learning behavior of the network instances in terms of cumulative regret and use financial metrics to compare with the following benchmarks:

- Capital Asset Pricing Model (CAPM) - [Fama and French \(2004\)](#);

- Portfolio Theory - [Markowitz \(1952\)](#);
- Risk Parity;
- Equal Weights;
- S&P Index.

**Table 3:** S&P Portfolio Optimization - Experiment Setup

Instance	Type	Layer 1 Param.	Reward Fn (Layer 1)	Layer 2 Param.	Reward Fn (Layer 2)	Size
1	<a href="#">3.3.1</a>	SW-UCB	Mean Return ( $w_f=100$ )	CADTS	Sharpe Index ( $w_f=60$ )	k=4
2	<a href="#">3.3.1</a>	ADTS	Mean Return ( $w_f=100$ )	CADTS	Sharpe Index ( $w_f=60$ )	k=4
3	<a href="#">3.3.2</a>	ADTS	Mean Return ( $w_f=100$ )	ADTS	Sharpe Index ( $w_f=60$ )	k=4
4	<a href="#">3.3.2</a>	ADTS	Mean Return ( $w_f=100$ )	ADTS	Sharpe Index ( $w_f=60$ )	k=10
5	<a href="#">3.3.2</a>	ADTS	Mean Return ( $w_f=100$ )	ADTS	Sharpe Index ( $w_f=60$ )	k=15

#### 4.4 Portfolio Selection Robustness Experiment

To verify the robustness of the Bandit Networks instances presented in Section [4.3](#) Table [3](#), we incrementally remove a rank of the nine best stocks in cumulative returns. For the number of the top removed stocks, we define the variable  $M$ . Table [4](#) displays the experiment setup containing the simulation steps and the list of removed stocks at each step. We investigate financial metrics such as Cumulative Returns, Sharpe Index, and Maximum Drawdown for each step and Bandit Network instance and compare the results with the S&P index and the CAPM portfolio model.

The results of these experiments yield insight into understanding the drift evolution of each studied Bandit Network instance and evaluate their dependencies to outlier performing stocks. In the next section, we present the results of three offered experiments in this study.

**Table 4:** S&P Portfolio Optimization Robustness - Experiment Setup

Step	Stocks Removed
1	-
2	[NVDA]
3	[NVDA, AVGO]
4	[NVDA, AVGO, TSLA]
5	[NVDA, AVGO, TSLA, LLY]
6	[NVDA, AVGO, TSLA, LLY, GE]
7	[NVDA, AVGO, TSLA, LLY, GE, AMD]
8	[NVDA, AVGO, TSLA, LLY, GE, AMD, AAPL]
9	[NVDA, AVGO, TSLA, LLY, GE, AMD, AAPL, XOM]
10	[NVDA, AVGO, TSLA, LLY, GE, AMD, AAPL, XOM, GOOG]

## 5 Results

The results of our experiments to assess the performance of the non-stationary bandits and the bandit network instances are presented in this section. As described in the previous section, we conducted three experiments: the stock picking experiment, the portfolio optimization experiment, and the portfolio robustness experiment.

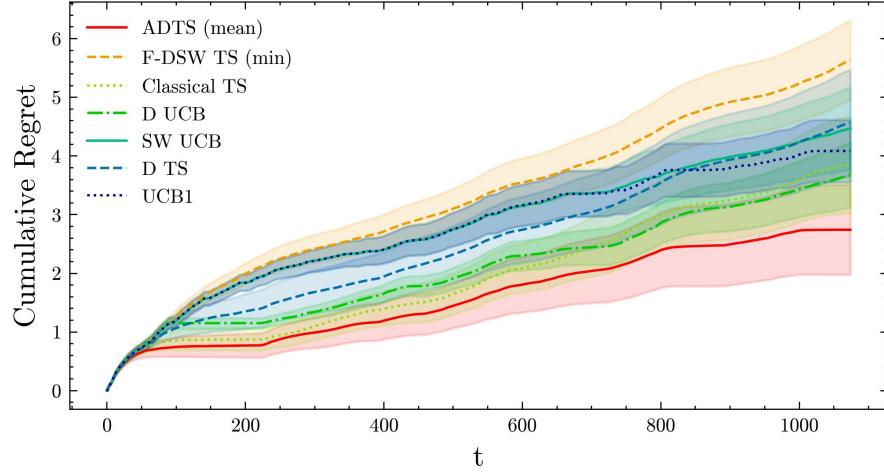
### 5.1 Stock Picking Experiment

The first set of experiment results is towards the S&P Stock Picking problem. The results are divided into three parts. In the first part, we analyze the learning characteristics of the ADTS against the bandit algorithms. Secondly, we obtain the financial metrics of each algorithm. Finally, we simulate the drift effect after applying shock in the top-performing stock of our S&P set.

#### 5.1.1 Regret Analysis

Figure 6 shows the cumulative regrets obtained according to the bandit algorithms present in Section 4.2 and Table 5 summarizes the results to help the reader to understand the differences.

The proposed ADTS is the one with the most prominent capability of detecting abrupt changes while presenting the lowest cumulative regret ( $3.2 \pm 0.7$ ) (as per the red line with credible intervals). The top three rankings are completed with D UCB ( $3.6 \pm 0.5$ ) and Classical TS ( $3.9 \pm 0.7$ ). It draws attention that F-DSW TS (Cavenaghi et al., 2021), the variant on which ADTS is based, presents the worst cumulative regret ( $5.6 \pm 0.6$ ) when applied to the S&P Stock Picking problem.



**Fig. 6:** Cumulative regret analysis, comparing the algorithms present in Table 2 (based on 30 simulations for each policy)

**Table 5:** Comparison of Cumulative Regrets with 95% Confidence Intervals (based on 30 simulations for each policy)

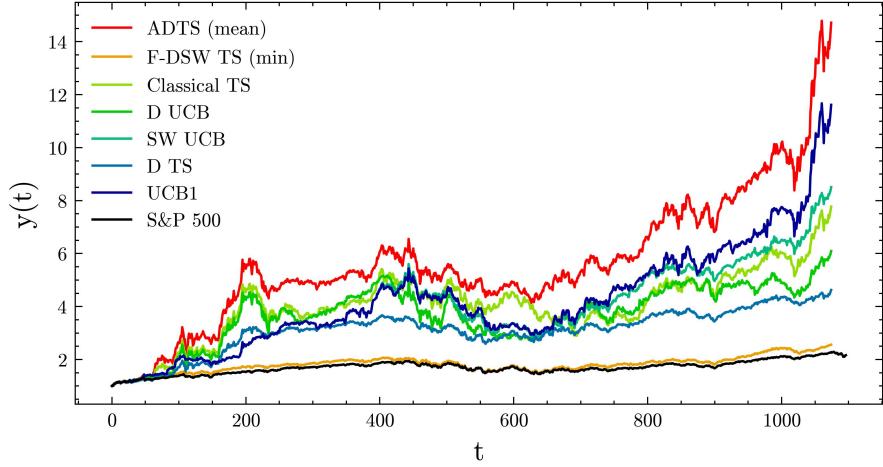
Algorithm	Mean Cumulative Regret (95% Conf.)
<b>ADTS (mean)</b>	$2.7 \pm 0.8$
D UCB	$3.7 \pm 0.6$
Classical TS	$3.8 \pm 0.8$
UCB1	$4.1 \pm 0.5$
SW UCB	$4.5 \pm 0.7$
D TS	$4.6 \pm 0.9$
F-DSW TS (min)	$5.6 \pm 0.7$

### 5.1.2 Financial Metrics Analysis

For a financial performance evaluation of the non-stationary Bandits policies, we investigate the following metrics: Return, Sharpe Ratio, Drawdown, Win Rate, and Sortino Ratio. Return quantifies profitability, while the Sharpe Ratio assesses risk-adjusted returns. Drawdown measures maximum loss, Win Rate indicates success frequency, and Sortino Ratio evaluates downside risk. These metrics collectively provide a comprehensive overview of a strategy's performance and risk profile. Results are stored in Table 6.

Figure 7 illustrates the cumulative returns obtained for each bandit algorithm in the experiment and the S&P Index. Not only does the ADTS present the highest stock picking capability, but it transforms it into considerably better returns compared to the other Bandit policies or the S&P 500 Index itself. When analyzing the Sharpe Ratio, UCB-1 is leading the metrics, followed by SW UCB and ADTS, in third. These three instances also stay at the top for the Sortino Ratio.

Compared to the S&P 500 Index, in terms of returns, all the bandits policies present superior performance than the S&P 500 Index. It is worth mentioning that all the policies can select one stock at a time, so maybe the higher drawdowns compared to the index, which is an aggregation of various stocks, are justified by this asymmetry.



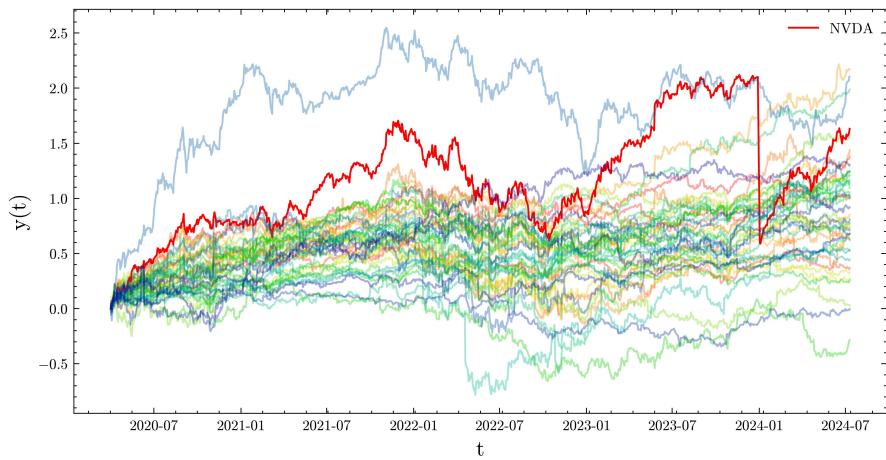
**Fig. 7:** Cumulative daily returns, comparing the algorithms present in Table 2 (based on 30 simulations for each policy)

**Table 6:** Policies financial performance metrics

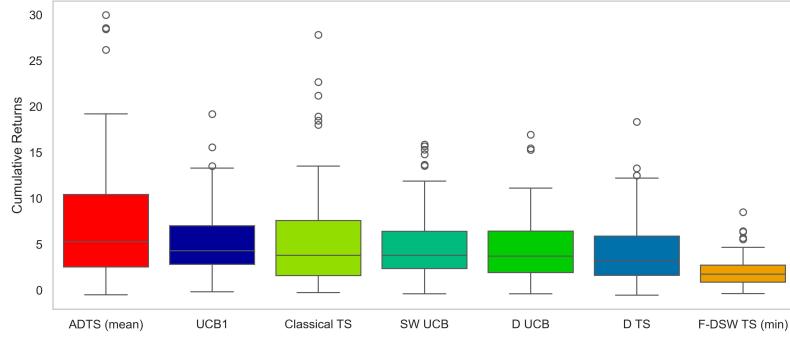
Policy	Total Return	Sharpe Ratio	Drawdown	Win Rate	Sortino Ratio
<b>ADTS (mean)</b>	<b>13.72</b>	1.76	2.42	0.55	0.16
UCB1	10.62	<b>1.91</b>	2.52	0.55	<b>0.19</b>
SW UCB	7.51	1.86	2.78	0.54	0.18
Classical TS	6.77	1.35	2.52	0.54	0.12
D UCB	5.10	1.21	2.45	0.55	0.11
D TS	3.63	1.51	1.08	0.54	0.13
F-DSW TS (min)	1.56	1.26	0.59	0.55	0.11
S&P 500	1.20	1.12	<b>0.49</b>	0.54	0.10

### 5.1.3 Drift Analysis

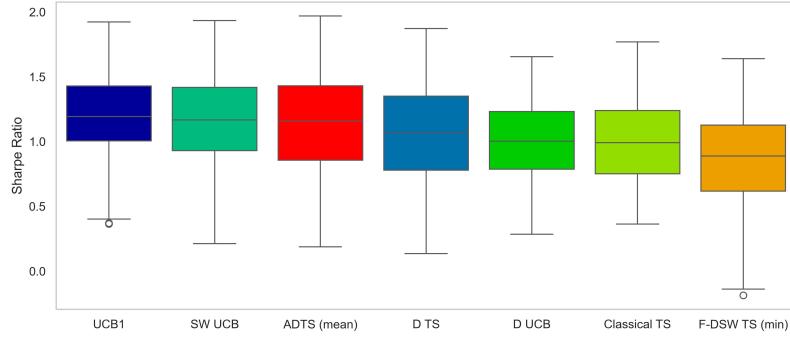
For analyzing the concept drift in the selected set of S&P stocks, we imposed an artificial shock to the best-performing ticker, the NVDA stock in January 2024. Figure 8 illustrates the change. With this transformation, we repeated the S&P Stock Picking problem for all the analyzed bandit algorithms, thirty (30) simulations for each policy. Results are shown in Figure 9. After the applied drift, the ADTS is the policy that presents the highest cumulative median returns (5.27), followed by UCB-1 (4.29), which represents an increase of 22.8%. In terms of the Sharpe Ratio UCB-1 is leading (1.20), closely followed by ADTS (1.17). Being the worst performer policy in cumulative returns and Sharpe Ratio, the F-DSW TS is leading the rank for presenting the best risk behavior, given its Drawdown of 1.19. On the other hand, our proposed policy, ADTS, had the second highest median Drawdown (3.70).



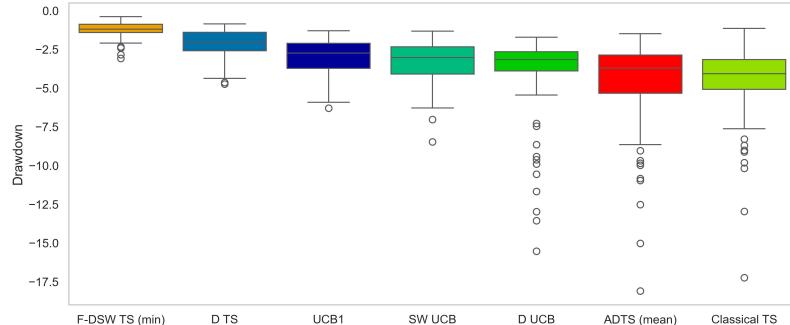
**Fig. 8:** Imposed drift to the best stock: NVDA



(a) Cumulative returns



(b) Sharpe Ratio



(c) Drawdown

**Fig. 9:** Drift analysis for the stock (30 simulations for each algorithm)

## 5.2 Portfolio Optimization Experiment

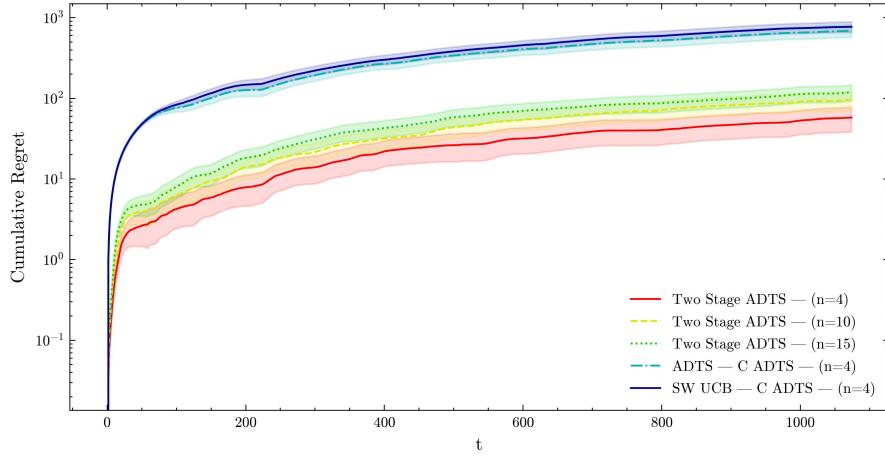
In the second experiment, we evaluate the performance of different Bandit Networks instances in the Portfolio Optimization Problem given a set of S&P stocks. The results are split into two parts. In the first part, we analyze the learning characteristics of the

Bandit Networks instances designed in Table 3. Finally, we investigate the financial metrics of each network instance and compare them to classical portfolio allocation models and the S&P Index.

### 5.2.1 Regret Analysis

Figure 10 shows the cumulative regrets in the log-scale for the y-axis obtained for the studied network instances. Table 7 summarizes the results to help the reader to understand the differences.

Comparing the regret results, the network instance 3 (Two-Layer ADTS, with  $n = 4$ ) stands out as the best learning configuration showing the lowest cumulative regret ( $57.5 \pm 19.5$ ). The top three are completed by the other two instances derived from Section 3.3.2, instances 4 and 5. There is a clear separation between the network instances proposed by Section 3.3.1. Instance 1 presents the worst mean cumulative regret value ( $767.2 \pm 113.6$ ), technically tied with instance 2 ( $685.3 \pm 122.5$ ).



**Fig. 10:** Cumulative regret analysis, comparing the network instances present in Table 3.

**Table 7:** Comparison of Cumulative Regrets with 95% Confidence Intervals (based on 30 simulations for each policy)

Bandit Network Instance	Mean Cumulative Regret (95% Conf.)
Two Layer ADTS (n=4)	$57.5 \pm 19.5$
Two Layer ADTS(n=10)	$96.4 \pm 25.3$
Two Layer ADTS (n=15)	$118.8 \pm 26.9$
ADTS   CADTS (n=4)	$685.3 \pm 122.5$
SW UCB   CADTS (n=4)	$767.2 \pm 113.6$

### 5.2.2 Financial Metrics Analysis

We move to analyze the financial metrics obtained for the bandit networks, comparing them with classical portfolio models. Results are stored in Table 8. Figure 11 illustrates the payoff chart of each of the bandit network instances results and the classical portfolio models and how they compare to the S&P index.

The Two-Stage ADTS ( $n = 4$ ) instance is the one with the most prominent results of cumulative returns (4.92), Sharpe and Sortino Ratios (1.59 and 0.14, respectively).

Sharpe Ratio value for Two-Stage ADTS ( $n = 15$ ) slightly loses to the best instance. By selecting fifteen stocks simultaneously, the instance diversifies risks, as suggested by the smallest drawdown metric of the bandit networks instances (0.55).

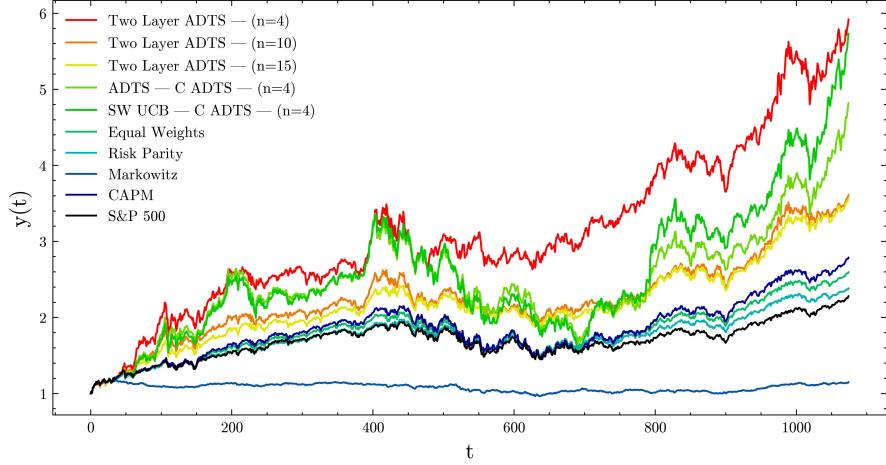
Contrary to the cumulative regrets suggestions, SW UCB | CADTS ( $n = 4$ ) and ADTS | CADTS ( $n = 4$ ) stands in second and third when taking the cumulative returns, although compromising their Sharpe Ratio having higher risks than the other three remaining instances.

Compared to the classical portfolio models, nominally CAPM, Equal Weights, Risk parity, Markovitz as well as the S&P 500 Index, in terms of returns, all the bandits network instances present superior performance. In this aspect, the cumulative returns of Two-Stage ADTS ( $n = 4$ ) is 168% higher than the CAPM, where the last is the best-performing classical model. The worst instance, Two-Stage ADTS ( $n = 15$ ), presents cumulative returns 42% higher than the best classical model, the CAPM, 2.55 against 1.79, respectively.

The pattern persists when it comes to the Sharpe Ratio. The best network instance in this criteria, Two-Stage ADTS ( $n = 4$ ), presents a Sharpe Ratio 20% higher than the best classical model, the Equal Weights, 1.59 against 1.32, respectively. The other instances also present superior values when compared to Equal Weights, except for ADTS | CADTS ( $n = 4$ ) (1.25), which marginally loses to equal weights and risk parity models.

**Table 8:** Policies financial performance metrics

Network Instance	Return	Sharpe	Drawdown	Win Rate	Sortino
Two Layer ADTS (n=4)	<b>4.92</b>	<b>1.59</b>	0.90	0.55	<b>0.14</b>
SW UCB   CADTS (n=4)	4.73	1.37	1.78	0.55	0.13
ADTS   CADTS (n=4)	3.82	1.30	1.70	0.56	0.12
Two Layer ADTS (n=10)	2.61	1.47	0.71	0.56	0.13
Two Layer ADTS (n=15)	2.55	1.58	0.55	0.56	0.14
CAPM	1.79	1.29	0.69	0.55	0.12
Equal Weights	1.59	1.32	0.59	0.56	0.12
Risk Parity	1.38	1.31	0.50	0.54	0.11
S&P 500	1.28	1.17	0.49	0.54	0.10
Markowitz	0.15	0.40	<b>0.22</b>	0.53	0.04



**Fig. 11:** Cumulative daily returns, comparing the network instances present in Table 3 (based on 30 simulations for each policy) and classical portfolio optimization models.

### 5.3 Portfolio Selection Robustness Experiment

To finish the set of experiments, we present the results depicted in the experiment setup presented in Section 4.4. To quantify how much each of the bandit network instances sustains their performance after removing the best stocks, we instantiate three types of financial metrics: i) Cumulative Returns (return), ii) Sharpe Ratio (return adjusted to risk), and iii) Drawdown (risk). Figures 12–14 show line plots of the three mentioned financial metrics as a function of the number of the best stocks ( $M$ ) shown in the experiment setup section. Table 9 consolidates the three analyzed metrics for each methodology, storing the values when  $M = 0, 4, 9$ . We compare the bandit network instances with the CAPM model and the S&P Index.

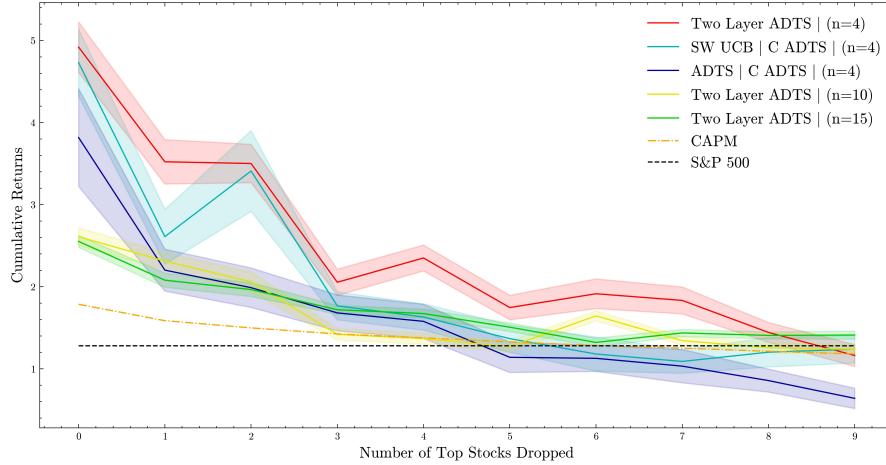
As shown in the table, the three instances with  $n = 4$  present higher drifts, which is logical due to the less diversification when compared to the instances with  $n = 10$  and  $n = 15$ .

For Cumulative Returns, the network that uses two layers of ADTS (red line) maintains the highest values until the number of the dropped best reaches eight. On the other hand, the networks that use CADTS in the last layer (Section 3.3.1) do not manage the same capability, as they start to lose to the  $n = 10$  and  $n = 15$  instances, and even the CAPM and S&P Index after removing six best stocks. The Two Layer ADTS ( $n = 15$ ), green line in the charts, demonstrates the highest bandit network capability of maintaining the cumulative returns after  $M = 9$ , preserving a value of 1.41, 19% higher than CAPM and 10% higher than the S&P Index. Overall, the analyzed bandit networks demonstrate higher cumulative return drift when compared to CAPM and S&P Index.

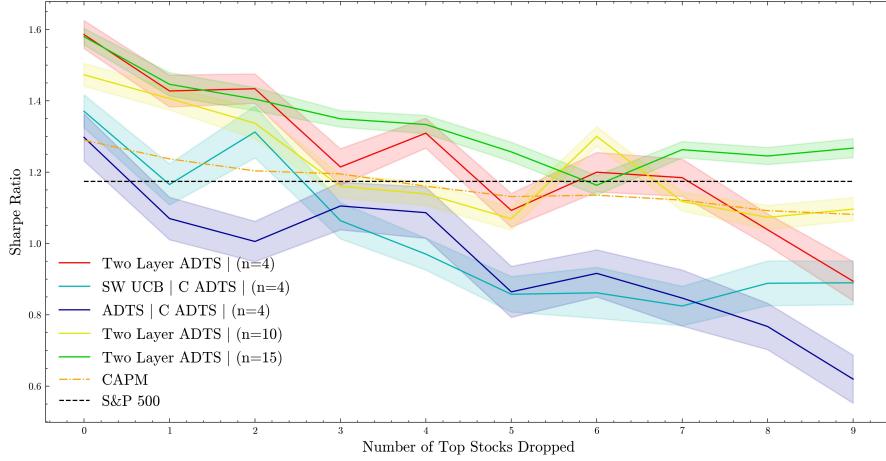
For the Sharpe Ratio, similar trends can be observed. The two-layer ADTS ( $n = 15$ ) showcases the lowest drift values amongst all the bandit network instances, demonstrating comparable stability to the CAPM model. In fact, after  $M = 9$  it

presents the highest value (1.27), 17% higher than the CAPM model, the second best. Similar to cumulative returns, in this criteria the higher drift values are also observed for the bandit network instances.

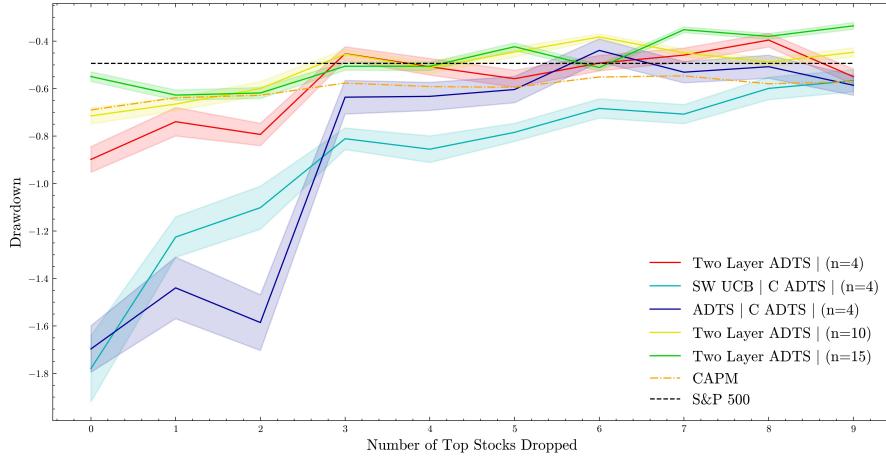
Finally, for the Drawdown metric, the two-layer ADTS ( $n = 15$ ) presents the top three less risky choices, including the CAPM and S&P Index. After the number of best stocks removed is equal or superior to seven, the instance presents the lowest value. This less risky behavior is certainly helping it to sustain the highest Sharpe Ratio, given the fact that its Cumulative Returns values are modest though stable, compared to the other instances.



**Fig. 12:** Cumulative Returns drift analysis of bandit networks instances, after incrementally removing the best stocks in cumulative returns given in Table 4.



**Fig. 13:** Sharpe Ratio drift analysis of bandit networks instances, after incrementally removing the best stocks in cumulative returns given in Table 4.



**Fig. 14:** Drawdown drift analysis of bandit networks instances, after incrementally removing the best stocks in cumulative returns given in Table 4.

## 6 Discussion

This section discusses the outcomes of our experiments on stock picking, portfolio optimization, and portfolio robustness. These experiments evaluate the performance of the newly introduced ADTS algorithm, as well as the novel concept of bandit networks

**Table 9:** Robustness Analysis

Network Instance	M=0	M=4	M=9	Total Drift (M=0 to M=9)
<b>Cumulative Return</b>				
Two Layer ADTS (n=4)	4.92	1.75	1.16	76.4%
SW UCB   CADTS (n=4)	4.73	1.37	1.24	73.8%
ADTS   CADTS (n=4)	3.82	1.14	0.64	83.2%
Two Layer ADTS (n=10)	2.61	1.27	1.24	52.5%
Two Layer ADTS (n=15)	2.55	1.51	1.41	44.7%
CAPM	1.79	1.33	1.18	33.9%
S&P 500	1.28	1.28	1.28	0.0%
<b>Sharpe Ratio</b>				
Two Layer ADTS (n=4)	1.59	1.09	0.89	43.6%
SW UCB   CADTS (n=4)	1.37	0.86	0.89	35.1%
ADTS   CADTS (n=4)	1.3	0.86	0.62	52.2%
Two Layer ADTS (n=10)	1.47	1.07	1.1	25.6%
Two Layer ADTS (n=15)	1.58	1.26	1.27	19.8%
CAPM	1.29	1.13	1.08	16.1%
S&P 500	1.17	1.17	1.17	0.0%

presented in this work. With this goal, we used a set of 44 stocks' historical daily returns of the S&P index, starting from April 2020 to July 2024.

The stock-picking experiments reveal the superior performance of the proposed ADTS algorithm. Regret analysis demonstrates that ADTS achieves the lowest cumulative regret, significantly outperforming other bandit algorithms, including its predecessor F-DSW TS. Financial metrics further support its efficacy, with ADTS yielding the highest returns among the tested algorithms, and also showing commendable results in terms of Sharpe and Sortino ratios. Notably, all bandit policies surpass the S&P 500 Index in returns, though they exhibit higher drawdowns, likely due to their single-stock selection constraint. Drift analysis, conducted by imposing a shock on the top-performing NVDA stock, underscores ADTS's robustness, maintaining high cumulative returns and a competitive Sharpe Ratio even under perturbations.

In portfolio optimization, the two-layer ADTS network with  $n = 4$  stocks stands out with the lowest cumulative regret and highest cumulative returns, Sharpe, and Sortino ratios. Compared to classical portfolio models like CAPM, Equal Weights, Risk Parity, Markowitz, and the S&P 500 Index, all bandit network instances exhibit superior performance. Notably, the cumulative returns of the two-layer ADTS ( $n = 4$ ) are 168% higher than CAPM, the best-performing classical model. Even the worst instance, two-layer ADTS ( $n = 15$ ), shows cumulative returns 42% higher than CAPM.

This trend continues with the Sharpe Ratio, where the two-layer ADTS ( $n = 4$ ) is 20% higher than Equal Weights, the best classical model in this regard. Other instances also surpass Equal Weights, except for ADTS | CADTS ( $n = 4$ ), which slightly trails behind Equal Weights and Risk Parity. While other network instances, particularly those combining ADTS and SW UCB, show higher cumulative regrets, they still outperform classical models in returns. The two-layer ADTS networks with higher  $n$  values (10 and 15) better diversify risks, as indicated by their lower drawdowns.

The robustness experiments corroborate the resilience of the two-layer ADTS networks, especially with higher  $n$  values. For cumulative returns, the two-layer ADTS

( $n = 4$ ) maintains the highest values until eight top-performing stocks are removed. In contrast, networks using CADTS in the last layer start losing to the  $n = 10$  and  $n = 15$  instances, and even the CAPM and S&P Index after removing six top stocks. The two-layer ADTS ( $n = 15$ ) demonstrates the highest robustness, maintaining cumulative returns 19% higher than CAPM and 10% higher than the S&P Index after removing the nine best-performing stocks.

For the Sharpe Ratio, the two-layer ADTS ( $n = 15$ ) shows the lowest drift values among all bandit network instances and is 17% higher than the CAPM model after  $M = 9$ . For the Drawdown metric, the two-layer ADTS ( $n = 15$ ) is among the top three least risky options, along with CAPM and the S&P Index, and presents the lowest drawdown when seven or more top stocks are removed. This low-risk behavior contributes to sustaining the highest Sharpe Ratio, highlighting the practical utility of the two-layer ADTS networks in maintaining portfolio performance amidst market fluctuations.

## 7 Conclusion

This work introduced and evaluated the ADTS algorithm and the concept of bandit networks through a series of experiments on stock picking, portfolio optimization, and portfolio robustness, using historical daily returns of 44 S&P 500 stocks from April 2020 to July 2024. The ADTS algorithm demonstrated superior performance, consistently achieving the lowest cumulative regret and highest returns, showcasing its effectiveness in both static and dynamic market conditions. The two-layer ADTS networks, particularly with  $n = 4$  and  $n = 15$ , exhibited remarkable robustness and risk-adjusted returns, outperforming classical models such as CAPM and Equal Weights.

The stock-picking experiments highlighted the ADTS's ability to maintain high returns and competitive Sharpe Ratios even under concept drift. In the portfolio optimization results, the two-layer ADTS networks efficiently learned and adapted, yielding superior cumulative returns and risk metrics. The robustness analysis further validated the stability of these networks, especially with higher  $n$  values, in maintaining performance amidst market fluctuations.

Future work could explore the application of ADTS and bandit networks to a broader range of financial instruments and market conditions. Additionally, enhancing the models to mitigate higher drawdowns observed in stock picking could further improve their practicality. Investigating the integration of alternative financial metrics and incorporating real-time adaptive mechanisms may also provide valuable insights for developing more resilient and adaptive financial decision-making tools.

**Acknowledgements.** This work was partially supported by grant 2022/01524-2, São Paulo Research Foundation (FAPESP).

## Declarations

- **Funding.** This study was supported by the Fundação de Amparo a Pesquisa do Estado de São Paulo (FAPESP - Grant 2022/01524-2).

- **Conflict of interest/Competing interests.** The authors have no relevant financial or non-financial interests to disclose.
- **Author's contribution.** All the authors contributed to the study's conception and design. Material preparation, data collection and analysis were performed by Gustavo Fonseca. The first draft of the manuscript was written by Gustavo Fonseca. Lucas Coelho and Paulo André Lima de Castro commented on previous versions of the manuscript. All the authors read and approved the final manuscript.

## References

- [1] Charpentier, A., Élie, R., Remlinger, C.: Reinforcement learning in economics and finance. *Computational Economics* **62**(1), 425–462 (2023) <https://doi.org/10.1007/s10614-021-10119-4>
- [2] Silva, N., Werneck, H., Silva, T., Pereira, A.C.M., Rocha, L.: Multi-Armed Bandits in Recommendation Systems: A survey of the state-of-the-art and future directions. *Expert Systems with Applications* **197**, 116669 (2022) <https://doi.org/10.1016/j.eswa.2022.116669>. Accessed 2024-03-16
- [3] Nakazato, S., Yang, B., Shimokawa, T.: Analyzing human search behavior when subjective returns are unobservable. *Computational Economics* **63**(5), 1921–1947 (2024) <https://doi.org/10.1007/s10614-023-10388-1>
- [4] Losada, D.E., Parapar, J., Barreiro, A.: Multi-armed bandits for adjudicating documents in pooling-based evaluation of information retrieval systems. *Information Processing & Management* **53**(5), 1005–1025 (2017) <https://doi.org/10.1016/j.ipm.2017.04.005>. Accessed 2024-03-16
- [5] Zhou, T., Wang, Y., Yan, L.L., Tan, Y.: Spoiled for Choice? Personalized Recommendation for Healthcare Decisions: A Multiarmed Bandit Approach. *Information Systems Research* **34**(4), 1493–1512 (2023) <https://doi.org/10.1287/isre.2022.1191>. Publisher: INFORMS. Accessed 2024-03-16
- [6] Bouneffouf, D., Rish, I., Aggarwal, C.: Survey on applications of multi-armed and contextual bandits. In: 2020 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8 (2020). <https://doi.org/10.1109/CEC48606.2020.9185782>
- [7] Castro, P.A.L., Annoni, R.: Towards autonomous investment analysts — helping people to make good investment decisions. In: 2016 Future Technologies Conference (FTC), pp. 74–80 (2016). <https://doi.org/10.1109/FTC.2016.7821592>
- [8] SBRANA, P.A. ATTILIO ; LIMA DE CASTRO: N-beats perceiver: A novel approach for robust cryptocurrency portfolio forecasting. *Computational Economics* **v. 1**, 101–136 (2023)
- [9] Castro, P.A.L., Parsons, S.: Modeling agent's preferences

based on prospect theory. In: MPREF@AAAI (2014).  
<https://api.semanticscholar.org/CorpusID:17027496>

- [10] Chen, Z., Ji, B., Liu, J., Mei, Y.: Multi-stage international portfolio selection with factor-based scenario tree generation. Computational Economics (2024) <https://doi.org/10.1007/s10614-024-10699-x>
- [11] Allesiardo, R., Féraud, R., Maillard, O.-A.: The Non-stationary Stochastic Multi-armed Bandit Problem. International Journal of Data Science and Analytics **3**(4), 267–283 (2017) <https://doi.org/10.1007/s41060-017-0050-5>
- [12] Lattimore, T., Szepesvári, C.: Bandit Algorithms. Cambridge University Press, ??? (2020)
- [13] Raj, V., Kalyani, S.: Taming non-stationary bandits: A bayesian approach. arXiv preprint arXiv:1707.09727 (2017) <https://doi.org/10.48550/arXiv.1707.09727> [stat.ML]. Submitted to NIPS 2017
- [14] Thompson, W.R.: On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. Biometrika **25**(3-4), 285–294 (1933)
- [15] Trovò, F., Paladino, S., Restelli, M., Gatti, N.: Sliding-window thompson sampling for non-stationary settings. J. Artif. Intell. Res. **68**, 311–364 (2020)
- [16] Cavenaghi, E., Sottocornola, G., Stella, F., Zanker, M.: Non-stationary multi-armed bandit: Empirical evaluation of a new concept drift-aware algorithm. Entropy (Basel) **23**(3), 380 (2021) <https://doi.org/10.3390/e23030380>. PMID: PMC8004723
- [17] Garivier, A., Moulines, E.: On upper-confidence bound policies for switching bandit problems. In: Kivinen, J., Szepesvári, C., Ukkonen, E., Zeugmann, T. (eds.) Algorithmic Learning Theory, pp. 174–188. Springer, Berlin, Heidelberg (2011)
- [18] Cao, Y., Wen, Z., Kveton, B., Xie, Y.: Nearly optimal adaptive procedure with change detection for piecewise-stationary bandit. In: Chaudhuri, K., Sugiyama, M. (eds.) Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 89, pp. 418–427. PMLR, ??? (2019). <https://proceedings.mlr.press/v89/cao19a.html>
- [19] Liu, F., Lee, J., Shroff, N.B.: A change-detection based framework for piecewise-stationary multi-armed bandit problem. In: AAAI Conference on Artificial Intelligence (2017). <https://api.semanticscholar.org/CorpusID:10480738>
- [20] Wang, H.: Large scale continuous-time mean-variance portfolio allocation via reinforcement learning. arXiv preprint arXiv:1907.11718 (2019)

- [21] Li, B., Hoi, S.C.: Online portfolio selection: A survey. ACM Computing Surveys (CSUR) **46**(3), 1–36 (2014)
- [22] Huo, X., Fu, F.: Risk-aware multi-armed bandit problem with application to portfolio selection. Royal Society open science **4**(11), 171377 (2017)
- [23] Shen, W., Wang, J., Jiang, Y.-G., Zha, H.: Portfolio choices with orthogonal bandit learning. In: Twenty-fourth International Joint Conference on Artificial Intelligence (2015)
- [24] Markowitz, H.: Portfolio selection. The Journal of Finance **7**(1), 77–91 (1952) <https://doi.org/10.2307/2975974>
- [25] Zhu, M., Zheng, X., Wang, Y., Li, Y., Liang, Q.: Adaptive portfolio by solving multi-armed bandit via thompson sampling. arXiv preprint arXiv:1911.05309 (2019)
- [26] Besbes, O., Gur, Y., Zeevi, A.: Stochastic multi-armed-bandit problem with non-stationary rewards. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, vol. 27. Curran Associates, Inc., ??? (2014)
- [27] Freitas Fonseca, G., Silva, L.C., Castro, P.A.: Addressing non-stationarity with relaxed f-discounted-sliding-window thompson sampling, 1–6 (2024) <https://doi.org/10.1109/COINS61597.2024.10622208>
- [28] Chen, W., Wang, Y., Yuan, Y.: Combinatorial multi-armed bandit: General framework and applications. In: Dasgupta, S., McAllester, D. (eds.) Proceedings of the 30th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 28, pp. 151–159. PMLR, Atlanta, Georgia, USA (2013). <https://proceedings.mlr.press/v28/chen13a.html>
- [29] Auer, P.: Using confidence bounds for exploitation-exploration trade-offs. Journal of Machine Learning Research **3**, 397–422 (2002)
- [30] Fama, E.F., French, K.R.: The Capital Asset Pricing Model: Theory and Evidence. Journal of Economic Perspectives **18**(3), 25–46 (2004)