Financial Asset Recommendation: Leveraging Multi-Armed Bandit Techniques

Bharat Khandelwal
Department of Data Science
George Washington University
Washington DC, USA

Amir Hossein Jafari
Department of Data Science
George Washington University
Washington DC, USA

***Abstract*—Financial asset recommendation (FAR) is challenging because financial markets are volatile, non-stationary, and only weakly predictable at short horizons. Classical recommender approaches such as popularity-based ranking or item–item collaborative filtering operate in a static setting and struggle to adapt to rapid changes in asset behavior. In this work, FAR was cast as a sequential decision problem and evaluated through the lens of multi-armed bandits (MABs). Using the FAR-Trans benchmark dataset, which contains anonymized retail investor transactions, asset metadata, and daily prices from 2018–2022, a contextual multi-armed bandit environment (CMAB) was constructed with weekly decision rounds and engineered features. The Asset-level context included the 7-day momentum, the 14-day volatility and market metadata, and a scenario expanded this to the customer's risk level. Across five controlled scenarios, the non-contextual UCB1 algorithm, the contextual LinUCB algorithm, a random baseline, and an item–item collaborative filtering recommender were compared using cumulative regret. The feature–reward analysis showed an extremely weak correlation between engineered features and forward-looking weekly returns, and the empirical results reflect this: regret grows almost linearly in all settings, and LinUCB behaves similarly to UCB1. Adding a customer risk level does not improve performance, and collaborative filtering underperforms bandit methods. These findings highlight both the limitations of short-horizon features for FAR and the usefulness of regret as a diagnostic metric for sequential financial recommendation.**

***Index Terms*—Financial asset recommendation, multi-armed bandits, contextual bandits, LinUCB, UCB1, FAR-Trans, cumulative regret.**

I. Introduction

Financial technology has made retail investing more accessible and increased interest in automated financial asset recommendation (FAR) tools. Unlike e-commerce or media recommendation, FAR operates in a dynamic, noisy environment where asset values evolve under changing macroeconomic conditions, market regimes, and shocks such as the COVID-19 pandemic and the Russia–Ukraine war. Recommendation systems in this setting must adapt to unstable asset trends rather than rely solely on static transaction patterns.

Sanz-Cruzado *et al.* introduced the FAR-Trans dataset, a benchmark containing anonymized investor transactions, asset metadata, and historical prices from a European financial institution over 2018–2022 [1]. Their study evaluated eleven algorithms based on profitability and transaction data but did not consider multi-armed bandit (MAB) methods, even though bandits provide a natural framework for sequential decision-making under uncertainty.

Recent work has applied contextual bandits to portfolio optimisation, combinatorial allocation, hedging, and stock-level trading [3]–[6]. These studies demonstrate that bandit algorithms can balance exploration and exploitation in financial environments. However, most focus on trading or portfolio construction and rarely use real customer transaction data in a recommendation setting.

This study extends the FAR-Trans benchmark by:

- constructing a contextual multi-armed bandit environment with weekly decision rounds and engineered asset features, and
- evaluating contextual (LinUCB) and non-contextual (UCB1) bandit algorithms, together with a random policy and item–item collaborative filtering, using cumulative regret as the primary metric.

The analysis focuses on a moderate-sized asset universe (68 or 253 assets) and weekly rounds derived from 253 Mondays. Most experiments use only asset-level features, with one scenario incorporating customer risk level. The results show that when engineered features are only weakly correlated with forward-looking returns, contextual bandits provide little advantage over non-contextual methods, and all algorithms suffer nearly linear regret.

II. Multi-Armed Bandits

A. Problem Setting

Multi-armed bandits provide a mathematical framework for sequential decision-making under uncertainty. Let $\mathcal{A} = \{1, \dots, K\}$ denote a finite set of arms. At each round $t = 1, \dots, T$, the agent selects an arm $a_t \in \mathcal{A}$ and receives a random reward $r_t \in \mathbb{R}$ drawn from an unknown distribution associated with that arm. Each arm a has an unknown expected reward $\mu_a = \mathbb{E}[r_t \mid a_t = a]$, and the optimal arm is

$$a^\star = \arg\max_{a \in \mathcal{A}} \mu_a. \tag{1}$$

Because the expectations μ_a are unknown, the agent must learn them while simultaneously trying to maximise its cumulative reward.

B. Regret as an Evaluation Metric

Performance is commonly measured using cumulative regret, which quantifies how much reward is lost by following a learning policy instead of always playing the optimal arm. The cumulative regret after T rounds is

$$R_T = T\mu_{a^\star} - \mathbb{E}\left[\sum_{t=1}^{T} r_t\right]. \tag{2}$$

A desirable algorithm achieves sublinear regret, $R_T = o(T)$, so that the average regret per round $R_T/T \to 0$ as T grows. In the financial setting considered here, regret reveals how well a recommendation policy tracks the best available asset over time and how strongly non-stationarity in returns affects learning.

C. Non-Contextual Bandits

In the non-contextual setting, the reward distribution of each arm depends only on the arm identity and is assumed stationary. The agent bases its decisions at time t on the history

$$\mathcal{H}_t = \{(a_1, r_1), \ldots, (a_{t-1}, r_{t-1})\}. \tag{3}$$

Several classic algorithms address the exploration–exploitation trade-off, including ϵ-greedy, Thompson Sampling, and Upper Confidence Bound (UCB) methods [8]. In this work, UCB1 is used as the non-contextual baseline.

Let $\hat{\mu}_a(t)$ denote the empirical mean reward of arm a after $t-1$ rounds, and let $N_a(t)$ be the number of times arm a has been selected. UCB1 chooses

$$a_t = \arg\max_{a\in\mathcal{A}} \hat{\mu}_a(t) + \sqrt{\frac{2\log t}{N_a(t)}}. \tag{4}$$

The confidence term encourages exploration of arms with few observations, while the empirical mean drives exploitation.

Classical stochastic bandit theory assumes that each arm's reward distribution is stationary over time. Financial time series, however, are often non-stationary due to changing regimes and volatility cycles, which can degrade the performance of non-contextual bandits [8], [9].

D. Contextual Bandits

Many practical decision problems involve additional side information, or context. In contextual bandits, the expected reward depends on both the chosen arm and an observed context describing the state of the environment, user, or item [2]. At each round t, the agent observes a context vector $x_t \in \mathbb{R}^d$, selects an arm a_t, and receives reward r_t with distribution dependent on (x_t, a_t).

A common model assumes a linear relationship between context and expected reward. In the arm-specific formulation,

$$\mathbb{E}[r_t \mid x_t, a_t = a] = x_t^\top \theta_a^\star, \tag{5}$$

where $\theta_a^\star \in \mathbb{R}^d$ is an unknown parameter vector for arm a. In many implementations, including LinUCB for this work, arm-specific context vectors $x_{t,a}$ and a shared parameter vector $\theta^\star$ are used:

$$\mathbb{E}[r_t \mid x_{t,a}, a] = x_{t,a}^\top \theta^\star. \tag{6}$$

E. LinUCB

LinUCB is a widely used contextual bandit algorithm that combines linear reward modelling with the optimism-in-the-face-of-uncertainty principle [7]. It maintains an estimate $\hat{\theta}_t$ of $\theta^\star$ and a design matrix A_t capturing feature covariance. For each arm a at round t with context $x_{t,a}$, LinUCB computes

$$p_{t,a} = x_{t,a}^\top \hat{\theta}_t + \alpha\sqrt{x_{t,a}^\top A_t^{-1} x_{t,a}}, \tag{7}$$

and selects

$$a_t = \arg\max_{a\in\mathcal{A}} p_{t,a}. \tag{8}$$

The first term estimates the expected reward, and the second term is an exploration bonus whose magnitude depends on the uncertainty in the feature space. After observing r_t, the algorithm updates A_t and $\hat{\theta}_t$ using ridge regression-style updates.

Contextual bandits benefit only when the features are informative about the reward. When the correlation between context and reward is weak, their advantage over non-contextual methods can vanish, an effect that appears clearly in the experiments.

III. System Model and Dataset

A. FAR-Trans Dataset

All experiments are based on the FAR-Trans dataset [1], which contains anonymized customer transactions, asset metadata, market information, and daily close prices from 2018 to 2022. The period includes significant turbulence driven by COVID-19 and geopolitical tensions [12], making short-term prediction particularly difficult.

The main components are:

- **Customer information:** identifiers, customer type (Mass, Premium, Professional, Legal Entity), risk level (Conservative, Income, Balanced, Aggressive and predicted variants), investment capacity brackets and predictions, and questionnaire timestamps.
- **Asset information:** ISIN, asset names, asset category (stock, bond, mutual fund), subcategory, sector, industry, market identifier, and timestamps.
- **Market information:** market and exchange identifiers, trading days and hours, country, and market class.
- **Close prices:** daily close prices in euros per ISIN.
- **Transactions:** customer and ISIN identifiers, transaction type, timestamp, units, total value, channel (online, phone, branch), and market ID.

Although FAR-Trans was not specifically designed for bandit algorithms, its structure allows the construction of decision rounds, context features, and reward signals for a bandit-based FAR environment.

B. Preprocessing Pipeline

The raw data are not directly suitable for contextual or non-contextual bandits. A multi-step preprocessing pipeline converts them into weekly snapshots with engineered features and rewards:

1) **Asset filtering and stability selection:** Assets with incomplete or irregular price histories are removed. Depending on the scenario, a stable universe of 68 or 253 assets is retained.
2) **Weekly snapshot construction:** All dates are aligned to Mondays. Each Monday defines a decision round and is assigned a snapshot identifier.
3) **Customer sampling:** For each snapshot, a fixed number of customers (500, 1000, or 3000) is sampled, depending on the scenario. Most scenarios use only asset-side features; one scenario adds customer risk level.
4) **Feature engineering:** For each asset in each snapshot, short-term market dynamics are encoded through 7-day momentum and 14-day volatility:

$$\text{momentum}_{7d} = \frac{P_t - P_{t-7}}{P_{t-7}}, \tag{9}$$

$$\text{volatility}_{14d} = \sqrt{\frac{1}{14}\sum_{i=1}^{14}(r_i - \bar{r})^2}, \tag{10}$$

where P_t is the price and r_i are daily returns. Additional categorical features include asset category and market identifier. In one scenario, customer risk level is appended as an extra contextual feature.
5) **Reward computation:** For each asset, the forward-looking weekly reward is defined as

$$r_{t+1} = \frac{P_{t+7} - P_t}{P_t}, \tag{11}$$

which allows direct comparison across algorithms.
6) **Final dataset construction:** The resulting CMAB-ready dataset contains one row per (timestamp, customer, asset) triplet, with feature vectors and reward labels. This forms the basis for all experiments.

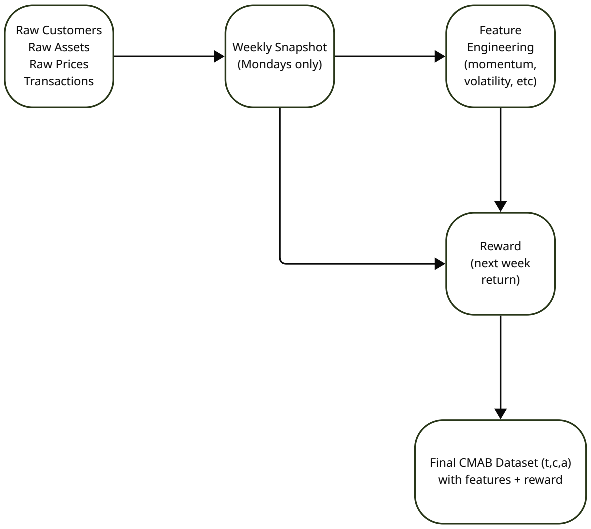

Fig. 1. Preprocessing pipeline and CMAB environment construction.

C. Feature–Reward Correlation

A Pearson correlation analysis between the engineered features and the forward-looking weekly reward shows that all features, including 7-day momentum, 14-day volatility, and country or market encodings, have extremely low correlation with realized returns. The largest absolute correlation is roughly 0.04, indicating very limited linear predictive signal.

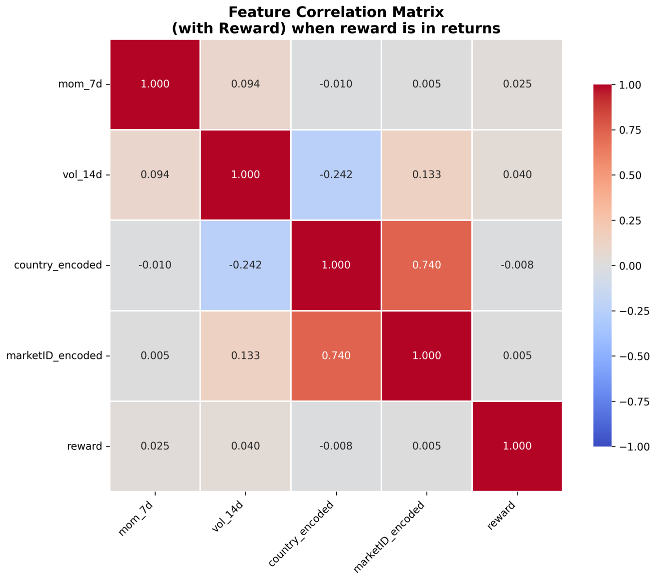

Fig. 2. Correlation between weekly asset features and forward-looking reward.

This weak association helps explain the behaviour observed in the regret curves: contextual bandits cannot exploit context effectively when features do not predict rewards [2].

IV. Experimental Setup

A. Bandit Algorithms

Four recommendation strategies are compared:

- **Random:** arms are sampled uniformly from the asset universe, providing a lower bound on performance.
- **Item–item collaborative filtering (CF):** a static recommender that uses customer co-purchase behaviour to rank assets. Cosine similarity between assets i and j is computed as

$$\text{sim}(i,j) = \frac{|U(i) \cap U(j)|}{\sqrt{|U(i)|\,|U(j)|}}, \tag{12}$$

where $U(i)$ is the set of customers who bought asset i. For a customer with history H_u, asset scores are

$$s_u(a) = \sum_{h \in H_u} \text{sim}(a,h). \tag{13}$$

In the bandit evaluation, CF proposes the top-ranked asset.
- **UCB1:** the non-contextual bandit algorithm defined in (4).
- **LinUCB:** the contextual bandit algorithm defined in (7). In Scenarios 1–4, only asset-side features are used; in

Scenario 5, customer risk level is appended to the feature vector.

All algorithms share the same sequence of decision rounds, asset universe, and reward definition to ensure fair comparison. Cumulative regret is evaluated over the full 2018–2022 horizon.

B. *Experimental Scenarios*

Five scenarios are designed to study how asset universe size, customer volume, and availability of context affect performance:

1) 68 assets, 500 customers, asset features only;
2) 68 assets, 3000 customers, asset features only;
3) 253 assets, 500 customers, asset features only;
4) 253 assets, 1000 customers, asset features only;
5) 253 assets, 1000 customers, asset features plus customer risk level.

TABLE I
SUMMARY OF EXPERIMENTAL SCENARIOS

Scenario	Assets	Customers	Features
1	68	500	Asset-only
2	68	3000	Asset-only
3	253	500	Asset-only
4	253	1000	Asset-only
5	253	1000	Asset + risk level

Increasing the number of assets expands the exploration space, while varying the number of customers per snapshot changes the number of recommendations made at each decision round. Scenario 5 isolates the impact of adding customer risk level as a contextual feature for LinUCB.

V. RESULTS

A. *Asset-Only Scenarios*

Figures 3–6 depict the cumulative regret trajectories for Random, CF, UCB1 and LinUCB in Scenarios 1–4. In all asset-only settings, regret grows almost linearly over time for every algorithm, indicating the difficulty of exploiting a stable best arm in this non-stationary environment.

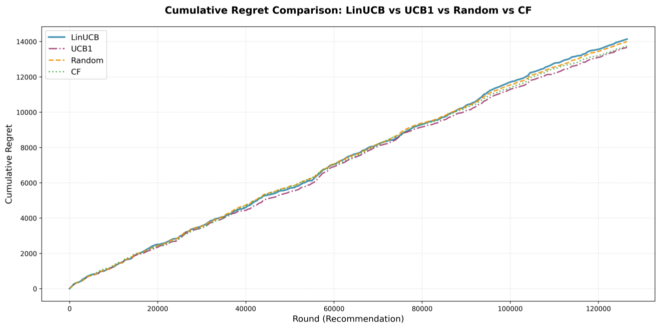

Fig. 3. Cumulative regret in Scenario 1 (68 assets, 500 customers).

With 68 assets and 500 customers (Scenario 1), UCB1 achieves slightly lower regret than LinUCB, while CF and Random lag behind. Increasing the number of customers to 3000 (Scenario 2) smooths the curves but does not change their ordering; LinUCB and UCB1 remain close, and static CF continues to underperform the bandit methods.

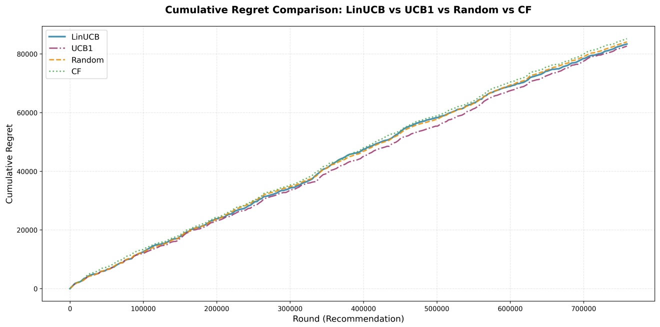

Fig. 4. Cumulative regret in Scenario 2 (68 assets, 3000 customers).

When the asset universe is expanded to 253 assets (Scenarios 3 and 4), all algorithms experience higher regret as the exploration space grows. LinUCB and UCB1 remain very close; in some runs LinUCB is slightly better, but no consistent dominance emerges. CF and Random again incur higher regret and do not close the gap.

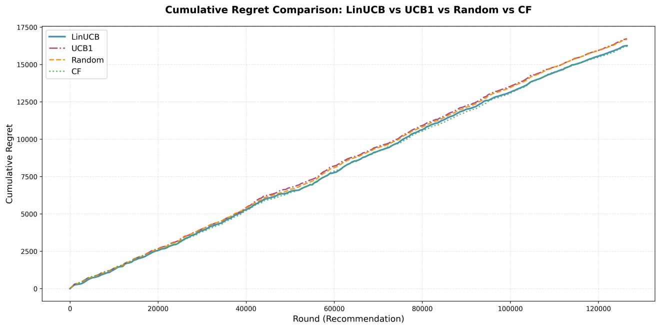

Fig. 5. Cumulative regret in Scenario 3 (253 assets, 500 customers).

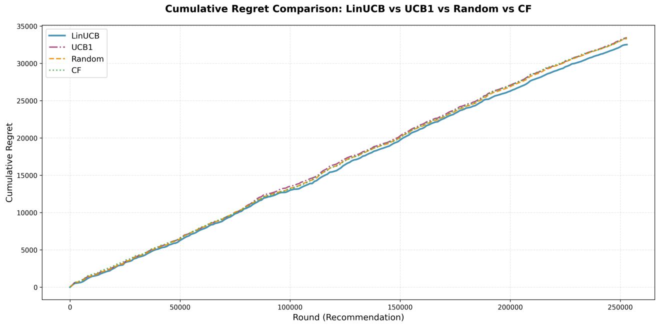

Fig. 6. Cumulative regret in Scenario 4 (253 assets, 1000 customers).

These results are consistent with the weak feature–reward correlations discussed earlier. When features do not reliably distinguish high-reward arms, contextual information offers little advantage, and both contextual and non-contextual bandits behave similarly.

B. Impact of Customer Risk Level

Scenario 5 adds customer risk level to the context used by LinUCB while keeping the asset universe and customer count identical to Scenario 4. The cumulative regret trajectory of LinUCB with risk-based context almost overlaps the asset-only LinUCB curve.

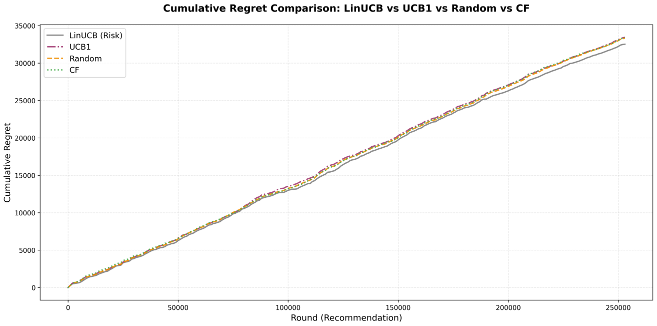

Fig. 7. Cumulative regret in Scenario 5 with risk-based context.

This suggests that, in this dataset, risk profiles capture preference differences but not short-horizon reward differences at the weekly level. Market movements dominate realized returns, and customer-level heterogeneity does not translate into more predictable weekly outcomes under the chosen reward definition.

C. Final Regret Comparison

Table II summarizes the final cumulative regret for each algorithm in all scenarios. UCB1 and LinUCB consistently achieve the lowest regret and remain very close to each other. CF and Random generally perform worse. Introducing customer risk level in Scenario 5 does not materially change LinUCB's performance.

TABLE II
FINAL CUMULATIVE REGRET ACROSS SCENARIOS

Scen.	Rand.	CF	UCB1	LinUCB	LinUCB-R
1	13991.18	13743.74	13664.14	14134.75	–
2	84052.80	85153.74	82649.09	83347.65	–
3	16682.10	16220.62	16726.83	16262.33	–
4	33332.86	33404.51	33453.66	32524.67	–
5	33332.86	33404.59	33453.66	–	32524.67

Overall, the experiments show that when engineered features have very limited predictive power for future returns, bandit algorithms cannot achieve sublinear regret, and contextual bandits do not outperform non-contextual ones. The regret analysis exposes the underlying non-stationarity and lack of exploitable structure in short-term weekly returns for this period.

VI. CONCLUSION

This work evaluated contextual and non-contextual multi-armed bandit algorithms for financial asset recommendation using the FAR-Trans dataset. A contextual bandit environment was constructed with weekly decision rounds, 7-day momentum and 14-day volatility features, market metadata, and, in one scenario, customer risk level. UCB1, LinUCB, a random baseline, and item–item collaborative filtering were compared across five scenarios differing in asset universe size, customer volume, and feature sets.

The empirical results show that cumulative regret grows almost linearly in all scenarios, and LinUCB performs similarly to UCB1. These findings are consistent with the extremely weak correlation between the engineered features and forward-looking weekly returns and with the non-stationarity of financial markets over the 2018–2022 period. Adding customer risk level does not improve performance, and static collaborative filtering underperforms both bandit algorithms.

Despite these limitations, the study underlines the usefulness of regret as a diagnostic metric for sequential recommendation in finance and confirms that contextual bandits remain a promising approach when informative context is available [2], [4]. Future extensions could explore richer feature sets, alternative reward definitions, or hybrid bandit–reinforcement learning models [11] in order to better exploit structure in financial time series while retaining the sample efficiency and interpretability of bandit formulations.

REFERENCES

[1] J. Sanz-Cruzado *et al.*, "FAR-Trans: A Benchmark Dataset for Financial Asset Recommendation," 2024. [Online]. Available: https://arxiv.org/pdf/2407.08692

[2] D. Bouneffouf and I. Rish, "A Survey on Practical Applications of Multi-Armed and Contextual Bandits," 2019. [Online]. Available: https://arxiv.org/pdf/1904.10040

[3] L. Cannelli, "Volatility regimes and non-stationary drifts pose a challenge for contextual bandits, as the reward distribution changes faster than the algorithm can adapt," *Quantitative Finance*, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S240591882300017X

[4] L. Ni, "Contextual Combinatorial Bandits for Financial Allocation," *Quantitative Finance*, 2023. [Online]. Available: https://arxiv.org/pdf/2407.00567

[5] D. Kar, "Stock-Level Trading Decisions Using Contextual Multi-Armed Bandits," 2024. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/3638530.3664145

[6] W. Chen and Z. Huang, "Contextual Bandits for Dynamic Portfolio Allocation," *Finance Research Letters*, 2024. [Online]. Available: https://researchportal.hkust.edu.hk/en/publications/enhancing-the-performance-of-bandit-based-hyperparameter-optimiza-2/

[7] L. Li, J. Langford, and R. E. Schapire, "A Contextual-Bandit Approach to Personalized News Article Recommendation," 2012. [Online]. Available: https://arxiv.org/pdf/1003.0146

[8] A. Slivkins, "Introduction to Multi-Armed Bandits," *Foundations and Trends in Machine Learning*, 2019. [Online]. Available: https://arxiv.org/pdf/1405.3316

[9] A. Goldenshluger and A. Zeevi, "A Linear Response Bandit Problem," 2014. [Online]. Available: https://pubsonline.informs.org/doi/epdf/10.1287/11-SSY032

[10] G. de Freitas Fonseca, L. Coelho e Silva, and P. A. Lima de Castro, "Improving Portfolio Optimization Results with Bandit Networks," 2025. [Online]. Available: https://doi.org/10.1007/s10614-025-11090-0

[11] Z. Zhang, S. Zohren, and S. Roberts, "Deep Learning for Portfolio Optimization," 2021. [Online]. Available: https://arxiv.org/pdf/2005.13665

[12] A. Baker and J. Wurgler, "Investor Sentiment and the Cross-Section of Stock Returns During Crise," 2006. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstractid=464843