A Comparative Study of Multi-Armed Bandit Algorithms for Dynamic Filter and Effect Recommendations on Tiktok

Jiacheng Zeng

College of Liberal Arts & Sciences, University of Illinois Urbana-Champaign, Champaign, Illinois, 61820, USA

Abstract. This study assesses how three benchmark Multi-Armed Bandit (MAB) methods—ε-greedy, Upper Confidence Bound (UCB), and Thompson sampling—perform in dynamically recommending filters and effects for short-video platforms. An experiment was configured using real user engagement metrics (likes, shares, views), enabling each algorithm to incrementally build reward estimates and adapt to evolving content trends. The study examines the dual impact of striking a diversity–satisfaction equilibrium on recommendation effect identification. Results suggest that moderate exploration strategies yield the highest effectiveness by enabling the discovery of emerging, underused effects without compromising user satisfaction. A hybrid MAB model is then proposed, in which exploration adjusters are reactive to performance metrics, allowing prompt responses to content trends while maintaining recommendation quality. Finally, practical aspects of deploying these models—such as latency, scalability, and fairness—are discussed, along with concrete recommendations for designing adaptive social media recommendation systems. By integrating conceptual and practical approaches, this study demonstrates how MAB algorithms can achieve both content discovery and reliability.

1 Introduction

With the booming development and increasing popularity of the Internet and short videos, TikTok, as the world's leading short video platform, has attracted a large number of users through interesting special effects and various beautiful moments shared by users. In the process of using the platform, users are increasingly looking forward to personalized, high-quality, and interesting content, and also want to use a variety of fun special effects to create videos. This requires the platform to accurately recommend suitable content to everyone, and at the same time, recommend new special effects. How to accurately recommend suitable content to users has become a key issue in improving user experience and platform competitiveness. The multi-armed bandit algorithm is a classic algorithm [1]. Its advantage is that it can not only constantly try new recommendation methods, but also use past data to find the most popular content, so as to capture the interest of users more

jz133@illinois.edu

quickly. This method has become a research hotspot in this field [2]. Now, several commonly used methods in TikTok, such as ε-Greedy, UCB, and Thompson sampling, have their own characteristics, some of which have fast exploration speed, and some are more conservative [3, 4]. These algorithms can not only improve the accuracy of recommendations but also change users' usage habits in continuous adjustments and increase users' dependence on the platform [5]. However, these algorithms are numerous, and there is still room for improvement. This report aims to explore the application of multi-armed bandit algorithms in TikTok's dynamic filtering algorithm and content recommendation, analyze the characteristics of different algorithms and their impact on users, and propose possible improvement strategies.

2 Related Work

During the literature review, existing research reveals the technical evolution and application challenges of MAB in self-media recommendation algorithms. Classic MAB algorithms (such as greedy strategies, UCB, and Thompson sampling) optimize recommendation effects through different "exploration-exploitation" balance mechanisms (Silva et al., 2022). For example, the UCB algorithm quantifies uncertainty through confidence intervals and enhances exploration capabilities in the cold start phase, while Thompson sampling dynamically adjusts strategies through Bayesian inference (Kuleshov & Precup, 2014). With the increasing complexity of TikTok scenarios, researchers have begun to integrate contextual information with the MAB framework, such as the contextual bandit model proposed by Pilani et al. (2021), which dynamically updates the recommendation policy through real-time user behavior data, significantly improving the short-term click-through rate. In response to the cold start and sparse data problems, Xu et al. (2021) innovatively combined Bandit-based Matrix Factorization (BanditMF), using implicit feedback to construct a user preference matrix, while exploring unknown interests through the bandit mechanism to achieve a balance between efficiency and accuracy.

At the enterprise and engineering level, the hybrid meta-learning method proposed by Cunha and Marchini (2024) further breaks through the limitations of single-objective optimization and achieves the scene adaptability of personalized recommendations through multi-objective and multi-user collaborative optimization. However, in the actual application of short video platforms such as TikTok, studies have pointed out that although its recommendation system relies on hybrid strategies (such as the combination of content collaborative filtering and MAB), it has the problem of insufficient optimization (Zhao, 2021; Zhang & Liu, 2021). For example, the situation where users quickly swipe to skip content is not taken into account, resulting in the recommendation algorithm's slow response to real-time changes in user preferences. This defect may exacerbate the "filter bubble" risk proposed by Tan and Yoon in 2024-over-reliance on homogeneous content recommendations will reduce long-term user engagement. In this regard, Hagar and Diakopoulos's (2023) empirical research shows that there is an "algorithmic indifference" phenomenon in the TikTok recommendation system, that is, the exposure rate of diverse news content is less than 12% of ordinary entertainment content, highlighting the necessity of a dynamic exploration mechanism design. It is worth noting that the Bandit selection of review properties (BanditProp) framework proposed by Wang et al. (2022) provides a new idea for the problem that recommendation algorithms always appear in a large amount of identical content through an attribute-level slot machine selection mechanism. In summary, there are two main research gaps in existing work:

First, algorithms have not kept up with the pace of content creation: TikTok adds a large number of special effects or stickers every day, and users use them to produce video content (such as shooting a creative video or recording a beautiful life). However, the

current algorithm is relatively slow, and the video shot with a certain special effect has a high playback volume, but the other effects are not recommended first. In addition, some popular special effects are pushed long after they are launched online, and the topic has passed its heat when users want to shoot videos for popular challenges.

Second, evaluation criteria do not grasp the real needs of users: traditional algorithms only rely on clicks to find good effects while ignoring the experience. For instance, TikTok always pushes popular special effects (users use those few over and over again, but other interesting and personalized ones are not recommended, and the content is highly homogenized). In addition, when users see a hot topic and want to shoot immediately, they are always half a beat late. At last, they do not consider the "sense of achievement in creation" (for example, using niche special effects to make a hit video, this value is not evaluated by the algorithm). These real pain points have not been solved by existing research.

From what's mentioned above, this research is going to compare which MAB will fit better in the use of a dynamic filter(using data to compare). Use each algorithm plot to analyze the difference between each algorithm. Suggestions for choosing each algorithm and any modifications for the TikTok context.

3 Methodology

3.1 Overview

To evaluate the efficacy of different strategies for dynamic filter and effect recommendations on TikTok, a series of experiments was designed to compare several foundational MAB algorithms [6-11]. This section details the dataset utilized, the formulation of the recommendation problem within the MAB framework, the specific algorithms implemented, the experimental procedure, and the metrics chosen for performance evaluation.

3.2 Dataset Preparation

The experiments leverage the "TikTok User Engagement Dataset", sourced from Kaggle repositories. This dataset encapsulates user interactions with TikTok videos, providing granular data suitable for simulating a recommendation environment. Key features relevant to this study include:

- User ID: Unique identifier for each user.
- Filter ID: Unique identifier for the filter or special effect applied to a video.
- Engagement Metrics: video_view_count, video_like_count, video_share_count associated with videos utilizing specific filters/effects.
- Special effects/Stickers: Categorical data on other visual elements used.
- Real-time Heat Index: A metric potentially indicating trending status at the time of interaction.

Before applying bandit algorithms, the dataset underwent preprocessing steps. The raw dataset contained on the order of tens of thousands of video records, encompassing a wide variety of filters/effects. There are the following preprocessing steps:

Step 1: Data cleaning. Removal of corrupt or incomplete records. Entries with missing crucial fields (e.g., missing filter ID or engagement info) were either dropped or imputed with conservative values (e.g., missing engagement set to 0) to avoid skewing results.

Step 2: De-duplication. Duplicate entries (if any user interactions were logged multiple times) were removed to prevent overweighting certain videos.

Step 3: Filtering low-interaction data. Pruning very low-engagement videos (e.g., videos with near-zero views or likes) under the assumption that they might represent either irrelevant outliers or content not widely served. This focuses the simulation on content with sufficient user feedback to inform learning.

Step 4: Normalization of engagement metrics. To combine metrics into a single reward, each engagement metric was normalized. For instance, view counts and like counts were each scaled to [0,1] by dividing by the maximum observed value or using a logarithmic scale to dampen outliers. This normalization prevents one metric (such as view count, which is typically much larger in magnitude than share count) from dominating the reward signal.

Step 5: Temporal sorting and windowing. Where applicable, sorting interactions chronologically (using timestamps or the provided heat index as a proxy for recency) to simulate a realistic sequence of events. In some experiments, employing a time-windowing approach, the dataset was segmented into time intervals (e.g., monthly batches) to examine how algorithms perform as trends evolve over time. This allowed testing the algorithms' adaptability by, for example, training them on past data and then evaluating future data segments, or by online simulation that moves through time. Table 1 shows several typical filters selected in this study and their corresponding user interaction indicator examples, which will be used for subsequent algorithm experiments.

Table 1 User interaction indicators of some TikTok filters

video_id	video_view _count	video_like_ count	video_share _count	video_com ment_count	filter_id
7017666017	343296	19425	241	0	F101
4014381136	140877	77355	19034	684	F102
9859838091	902185	97690	2858	329	F103
1866847991	437506	239954	34812	584	F104
7105231098	56167	34987	4110	152	F105
8972200955	336647	175546	62303	1857	F106
4958886992	750345	486192	193911	5446	F107
2270982263	547532	1072	50	11	F108
5235769692	24819	10160	1050	27	F109

3.3 Algorithms

To understand the fundamental trade-offs between exploration and exploitation in this context, the following standard MAB algorithms were implemented and compared, chosen for their distinct approaches [12]. The following parts are the introductions of each algorithm:

ε-Greedy: This algorithm explores with a fixed probabilityε and exploits (chooses the arm with the highest estimated reward so far) with probability $1-\varepsilon$. It is simple to implement but can be inefficient in exploration. The estimated reward $Q_t(a)$for an arm after t steps is typically the average reward observed from that arm.

UCB: The UCB algorithm is a classic multi-armed bandit method that solves the exploration-exploitation trade-off by constructing an "optimistic" confidence upper bound for each arm. The basic idea is that in each decision step, in addition to considering the current average estimated reward of the arm, an additional error term is added to the arm with low confidence (i.e., not fully explored), thereby encouraging the algorithm to give priority to arms with greater uncertainty, so as to discover the potential best arm faster. The

standard UCB algorithm (also known as UCB1) selects the arm according to the following rules in step t [13, 14]:

$$a_t = argmax\{Q_{t-1}(a) + \sqrt{\frac{2\ln t}{N_{t-1}(a)}}\} \quad (1)$$

where $Q_{t-1}(a)$ represents the average reward estimate of arm a at the end of step $t-1$; $N_{t-1}(a)$ represents the number of times arm a is selected at the end of step $t-1$; and t is the current total number of decision steps, which is used to construct the confidence upper bound.

In order to flexibly adjust the exploration intensity and adapt to different data distributions and application scenarios, this paper adopts a UCB variant with an adjustable coefficient c. Based on formula (1), this variant introduces c to control the weight of the exploration item. Its decision rule is:

$$a_t = argmax\{Q_{t-1}(a) + c\sqrt{\frac{\ln t}{N_{t-1}(a)}}\} \quad (2)$$

where c is the exploration coefficient, which is used to adjust the "optimism" level of the underexplored arm.

Thompson Sampling: Thompson Sampling is a multi-armed bandit algorithm based on Bayesian posterior sampling. It maintains a probability distribution of reward parameters for each arm, extracts a sample value from the current posterior distribution of each arm in each decision-making round, and selects the arm with the largest sample value to pull; the actual reward observed after pulling is used to correct the posterior distribution of the arm, thereby achieving a natural balance between exploring unknown arms and using the existing optimal arm. In general, if the reward is a binary event (such as a click or a like), the Beta distribution can be used with the Bernoulli likelihood for Bayesian modeling; if the reward is a continuous value (such as normalized views, likes, and shares), it is appropriate to assume that it follows a normal distribution and impose a normal prior distribution on the mean parameter. In this scenario, the normalized continuous interaction indicator is regarded as a reward signal, which is assumed to obey a normal distribution with μ as mean and σ^2 as variance, and a normal prior is used for μ. At each decision, a candidate mean $\tilde{\mu}$ is extracted from the posterior normal distribution of each arm, and the arm with the largest $\tilde{\mu}$ is selected for recommendation. The posterior distribution is then corrected through Bayesian updating based on the observed continuous rewards, thereby dynamically balancing exploration and exploitation.

3.4 Metrics

Average Reward: Average Reward measures the mean reward obtained per decision step over the course of the simulation. Formally, if r_t denotes the reward at step t, then after T steps:

$$Average\ Reward = \frac{1}{T}\sum_{t=1}^{T} r_t \quad (3)$$

This metric captures the algorithm's ability to maximize cumulative engagement (views, likes, shares) and reflects overall recommendation quality in both stationary and non-stationary environments.

Optimal Action Percentage: Optimal Action Percentage tracks the proportion of times the algorithm selects the true best arm—that is, the filter or effect with the highest underlying expected reward—up to each decision step. If a_t^* is the optimal arm at step t and $\hat{a}_t$ is the arm chosen, then over T steps:

$$Optimal\ Action = \frac{1}{T}\sum_{t=1}^{T} I\{\hat{a_t} = a_t^*\} \times 100\% \quad (4)$$

4 Results

4.1 Stable Environment Experiment

In a stable scenario where the true reward distribution of all arms remains unchanged, the performance of two types of algorithms, ε-Greedy (ε=0, 0.01, 0.1) and UCB1 variant (c=2), is compared. The specific results are shown in Figures 1 and 2. As shown in Figure 1, the pure greedy strategy (ε=0, blue) quickly locks on the first high-reward arm in the first few dozen steps, and then the average reward stabilizes at about 1.3, indicating that the lack of exploration causes it to fall into a suboptimal arm. ε=0.01 (orange) uses extremely low probability random exploration. Although there is a slight delay in early convergence, it surpasses pure greed and stabilizes at a level close to 2.0 after about 500 steps, indicating that a small amount of exploration can find a better arm while ensuring high utilization. ε=0.1 (green) has a stronger exploration intensity and approaches the optimal average reward (about 2.05) in only about 200 steps. The curve fluctuates greatly in the early stage but then becomes smooth, proving that a higher exploration rate can accelerate the discovery of the optimal arm. The UCB1 variant (c=2, red) reaches a steady state in about 300-400 steps by relying on the confidence upper bound mechanism, with the final average reward slightly exceeding 2.1, and the early fluctuation is less than ε=0.1, showing better stability and convergence speed. In general, moderate or even strong exploration strategies can quickly locate and continuously use the optimal arm in a stable environment, thereby obtaining higher long-term cumulative benefits; while strategies that completely deprive exploration are prone to local optimality and are difficult to tap the potential of long-tail arms.

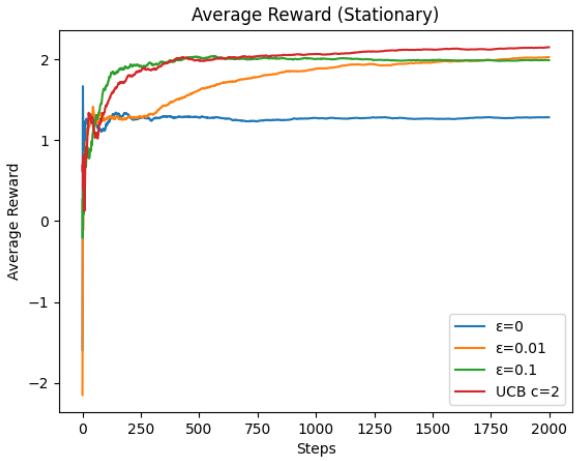

Figure 1 Average Reward (Stationary) (Picture credit: Original)

As shown in Figure 2, the optimal action selection ratio of each algorithm in a stable environment changes with the number of steps as follows: the pure greedy strategy (ε=0, blue) is unable to perform any exploration, so the optimal action ratio remains at 0; ε=0.01 (orange) almost does not select the optimal arm in the first 1500 steps, and only starts to increase slowly after the cumulative number of explorations reaches a certain threshold, and the optimal action ratio is about 28% at the 2000th step; ε=0.1 (green) completes the warm-up in a very short early stage, and the optimal action ratio has exceeded 80% at about the 200th step, and continues to rise steadily in the future, eventually tending to more than 90%. This result shows that a high exploration rate can significantly accelerate the discovery and utilization of the optimal arm, while a very low exploration rate can ensure early stability, but it takes longer to correct the early misjudgment of the suboptimal arm.

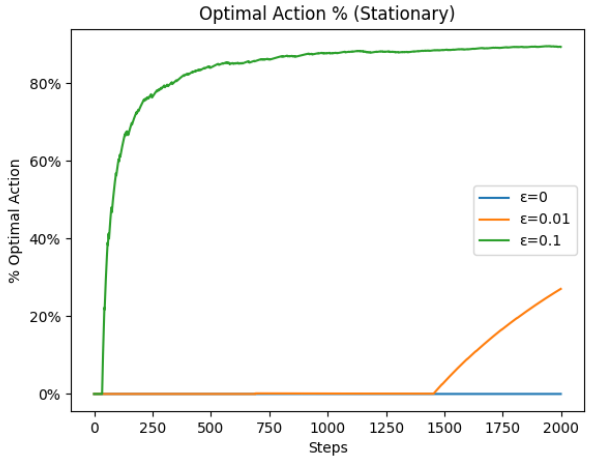

Figure 2 Optimal Action (Stationary) (Picture credit: Original)

4.2 Non-stationary Environment Experiment

In order to examine the adaptability of the same ε-Greedy algorithm in the scenario where the reward distribution drifts over time, this section fixes ε=0.1, uses the sample average update method and the fixed step size update method (α=0.1) to estimate the action value, and compares the performance of the two in a non-stationary environment. The specific results are shown in Figures 3 and 4.

As shown in Figure 3, in a non-stationary environment, the average reward curve of the fixed step update method (α = 0.1) is slightly higher than that of the sample average method. Both methods have large fluctuations in the first few hundred steps, but the fixed step method responds more quickly to reward drift and has maintained a slight lead since about 1,000 steps, reaching an average reward of about 1.05 at 10,000 steps. The sample average method has a slower average reward growth due to the excessive weight of historical data and a delayed response to the latest changes.

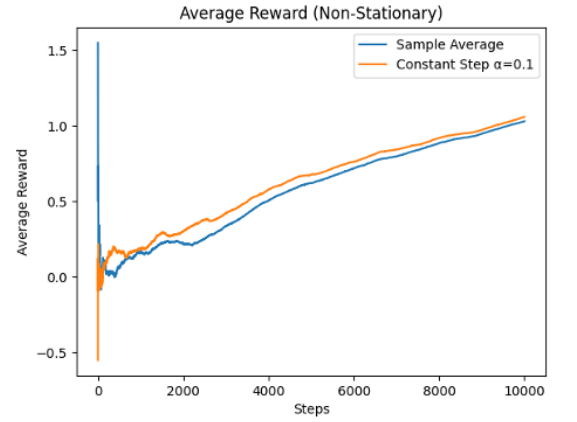

Figure 3 Average Reward (Non-Stationary) (Picture credit: Original)

As shown in Figure 4, the optimal action selection ratios of the two update methods also experienced violent fluctuations in the non-stationary scenario and then gradually recovered. The fixed step size method climbed to about 60%–65% in the first 500 steps and remained above 75%; the sample average method required more than 2,000 steps to break through 40%, and finally reached about 76% at 10,000 steps. This result shows that fixed step size updates are more conducive to the algorithm to quickly track and utilize dynamically changing optimal arms.

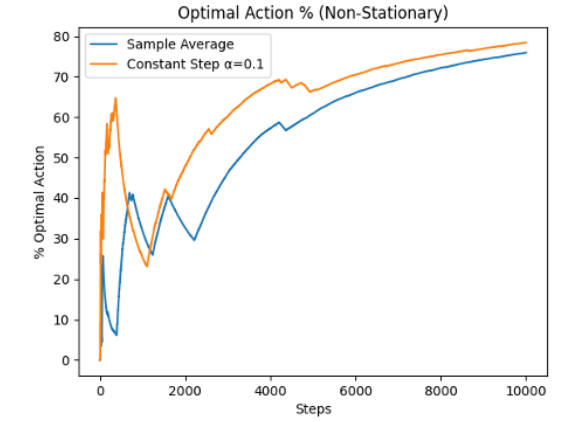

Figure 4 Optimal Action (Non-Stationary) (Picture credit: Original)

In summary, in experiments in stable environments, higher exploration rates (ε=0.1) and adaptive confidence bounds (UCB1 c=2) can quickly locate the optimal arm within hundreds of steps and maintain a high average reward, while extremely low or no exploration (ε=0, ε=0.01) is prone to being trapped in the suboptimal arm and it is difficult to obtain higher long-term benefits; in experiments in non-stationary environments, the fixed step size update method (α=0.1) has a faster response to reward drift than the sample average method, and it shows a steady lead in both the average reward level and the optimal action selection ratio, indicating that in dynamically changing scenarios, introducing update rules that give higher weight to recent information can significantly improve the algorithm's adaptability and tracking effects.

5 Conclusion

This study aims to meet the dynamic recommendation needs of filters and special effects in short video platforms. Based on real user engagement indicators (views, likes, and shares), an MAB simulation environment is constructed. The performance differences of three representative algorithms - ε-Greedy, UCB1 variants, and Thompson Sampling - in stable and non-stationary scenarios are compared, and the adaptability of different value update strategies (sample average and fixed step size) to changing trends is examined. The main conclusions are as follows:

First, in a stable environment, the strategy of completely depriving exploration (ε=0) converges to the suboptimal arm too early, and the final average return is limited; although a small amount of exploration (ε=0.01) converges slightly slower, it can discover a better arm; both high exploration (ε=0.1) and UCB1 (c=2) can quickly locate the optimal arm within hundreds of steps and maintain the highest long-term average reward and optimal action selection rate. Among them, UCB1 has smaller fluctuations in the early stage and slightly better stability.

Second, in a non-stationary environment, the fixed step size update (α=0.1) has a faster response to reward drift than the traditional sample average, and the average reward and the optimal arm selection ratio are both leading, reflecting the advantages of the update rule that gives higher weight to the latest information in dynamic scenarios.

Based on the above findings, a moderate or even strong exploration mechanism and an update strategy that responds agilely to recent data are the keys to achieving a balance between "content discovery-robust recommendation" for filters and special effects. For actual deployment, it is recommended to introduce an adaptively adjustable hybrid MAB framework under the premise of ensuring online inference latency and system scalability, and dynamically adjust the exploration parameters by real-time monitoring of reward trends to balance novelty, fairness, and user satisfaction, so as to provide theoretical and practical references for building the next generation of user-centric recommendation systems for social short video platforms.

References

1. Silva, N., Werneck, H., Silva, T., Pereira, A.C., and Rocha, L.: 'Multi-armed bandits in recommendation systems: A survey of the state-of-the-art and future directions', Expert Syst. Appl., 2022, 197, pp. 116669
2. Zhao, Z.: 'Analysis on the "Douyin (TikTok) Mania" phenomenon based on recommendation algorithms'. Proc. E3S Web of Conferences, 2021, 235, pp. 03029
3. Hagar, N., and Diakopoulos, N.: 'Algorithmic indifference: The dearth of news recommendations on TikTok', New Media Soc., 2023, pp. 14614448231192964
4. Zhang, M., and Liu, Y.: 'A commentary of TikTok recommendation algorithms in MIT Technology Review 2021', Fundam. Res., 2021, 1, (6), pp. 846-847
5. Pilani, A., Mathur, K., Agrawald, H., Chandola, D., Tikkiwal, V.A., and Kumar, A.: 'Contextual Bandit Approach-based Recommendation System for Personalized Web-based Services', Appl. Artif. Intell., 2021, 35, (7), pp. 489–504
6. Yan, C., Han, H., Wang, Z., and Zhang, Y.: 'Two-phase multi-armed bandit for online recommendation'. Proc. IEEE 8th Int. Conf. Data Science and Advanced Analytics (DSAA), Oct. 2021, pp. 1-8
7. Zhao, J.: 'Comparison of Multi-Armed Bandit Algorithms in Advertising Recommendation Systems', Appl. Comput. Eng., 2024, 83, pp. 62-71

8. Xu, S.: 'BanditMF: Multi-Armed Bandit Based Matrix Factorization Recommender System', arXiv preprint arXiv:2106.10898, 2021
9. Cunha, T., and Marchini, A.: 'A Hybrid Meta-Learning and Multi-Armed Bandit Approach for Context-Specific Multi-Objective Recommendation Optimization', arXiv preprint arXiv:2409.08752, 2024
10. Wang, X., Ounis, I., and Macdonald, C.: 'BanditProp: Bandit selection of review properties for effective recommendation', ACM Trans. Web, 2022, 16, (4), pp. 1-19
11. Tan, Y., and Yoon, S.: 'Testing the effects of personalized recommendation service, filter bubble and big data attitude on continued use of TikTok', Asia Pac. J. Mark. Logist., 2024
12. Chen, Y., and Huang, J.: 'Effective content recommendation in new media: Leveraging algorithmic approaches', IEEE Access, 2024
13. [de Campos Silva, N.: 'Active learning in contextual bandits: handling the uncertainty about the user's preferences in interactive recommendation systems', 2023
14. Cañamares, R., Redondo, M., and Castells, P.: 'Multi-armed recommender system bandit ensembles'. Proc. 13th ACM Conf. Recommender Systems, Sept. 2019, pp. 432-436