Proceedings of CONF-CIAP 2026 Symposium: Applied Mathematics and Statistics
DOI: 10.54254/2753-8818/2026.CH30042

An Empirical Comparison of Bayesian LinUCB, UCB, and Thompson Sampling for Recommendation on MovieLens

Jingyun Wang

School of Big Data and Software Engineering, Chongqing University, Chongqing, China
cquwangjy7@gmail.com

Abstract. T Recommender systems have evolved into core business hubs, with approximately 35% of Amazon's revenue stemming from recommendation-guided behaviors. This study conducts a systematic comparative analysis of three multi-armed bandit algorithms—Bayesian Linear Upper Confidence Bound (Bayesian LinUCB), Upper Confidence Bound (UCB), and Thompson Sampling—using the MovieLens dataset. The research evaluates algorithm performance across three key dimensions: cumulative regret, optimal arm selection frequency, and regret rate. Experimental variables are strictly controlled with consistent parameters, including decision steps and data division ratios to eliminate confounding factors. Results reveal significant performance differences among the algorithms within the limited experimental steps on the MovieLens dataset. UCB demonstrates optimal performance with the lowest cumulative regret (817.93) and highest optimal arm selection frequency (0.9822), followed by Thompson Sampling with moderate performance (cumulative regret: 2776.36, selection frequency: 0.924). Bayesian LinUCB performs poorly across all metrics, showing the highest cumulative regret (34105.02), lowest selection frequency (0.1324), and a regret rate of approximately 1, indicating linear rather than sublinear growth. The sublinear growth characteristic exhibited by UCB and Thompson Sampling confirms their superior exploration-exploitation balance, while Bayesian LinUCB's linear growth pattern suggests inadequate adaptation to the MovieLens dataset scenario, highlighting the importance of algorithm-dataset compatibility in recommendation systems.

Keywords: Bayesian LinUCB, Upper Confidence Bound, Thompson Sampling, MovieLens, Cumulative Regret

1. Introduction

In the digital economy, recommender systems have evolved from "auxiliary tools" to core hubs connecting corporate services and user needs, with their value rooted in two key drivers. First, they exert a decisive impact on enterprises' economic benefits. As personalized services become a key competitive focus across industries, companies in fields such as e-commerce, streaming media, and advertising have invested huge resources in the research, development, and application of recommendation technologies. Per McKinsey's 2018 report, The E-Commerce Consumer Decision Journey, approximately 35% of Amazon's transaction conversions come from consumption

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2753-8818/2026.CH30042

behaviors guided by its recommendation system—this figure remained a key industry reference for evaluating the value of recommendation systems in 2023, underscoring their huge economic influence. Second, they are key solutions to address the contradiction between "explosive product growth" and "user information overload." Today's consumer market sees product iteration speed reach an all-time high; the situation of "choice redundancy" makes it difficult for users to fully grasp product details, and they are more likely to fall into "choice paralysis" during the decision-making process. Recommender systems, by accurately capturing user preferences and filtering key information, can effectively reduce users' decision-making costs and help them quickly find products that meet their needs. In short, the quality of recommender systems not only directly affects the market competitiveness and economic benefits of enterprises across industries but also influences their willingness to pay for recommendation technology research, development, and data resource investment. It also profoundly impacts users' initial perception and trust in products, shapes their subsequent purchase decisions, brand loyalty, and even long-term consumption habits, serving as crucial support for the mutual value realization between enterprises and users in the digital era.

2. Related work

Due to its better exploration and exploitation of sequential decision problems now, the multi-armed bandit (MAB) algorithm has become a more commonly researched algorithm. In early work, Auer et. al. (2002) were able to establish finite time regret bounds for algorithms like UCB1. In particular, these proved logarithmic regret over time and served as an established lower bound on MAB analysis [1]. Then MAB was extended to contextual settings LinUCB by Li et al. (2010), context (user/item features) was included for personalizing news recommendation; And in Linear Thompson Sampling (LinTS) by Agrawal and Goyal (2013), Thompson Sampling was applied on linear contextual bandit [2, 3].

In these past handful of years, the MAB study has become more realistic, tougher. For example, Wei & Srivastava (2021) looked into nonstationary bandits with changing reward distributions and introduced new kinds of sliding windows and discounted UCB, which did better than traditional bandit algorithms in nonstationary environments [4]. Zhu & Liu(2021) studied the case of distributed MABs where multiple agents collaborate over a network to effectively find the best arms, and it works even if the graph is disconnected [5].

More innovations, along with MAB, are up with modern machine learning paradigms. Qiu et al. (2022) combine contrastive self-supervised learning and UCB to improve the sample efficiency of online Reinforcement Learning (RL) over Linear Reinforcement Learning in Markov Decision Processes (LRMDPs) and Markov Games [6]. Zhu& Qiu (2024) propose BUCB-E, a Bayesian UCB-Explore, which is robust due to the use of priors in fixed-budget best-arm identification [7]. MABs with costly probes are explored by Elumar et al. (2024); arms with costs for information concerning arms, costly, are investigated; and a UCBp variant and Thompson sampling variant are created to aid decisions involving costs [8].

New fields also have more recent expansion of application: Wu et al. (2025) regard the path selection in the semantic communication network as a sleepy bandit problem and put forward a UCB-based path selection algorithm to enhance energy utilization and transmission accuracy [9]. Saday et al.(2025) propose the Byzantine proof MAB algorithm Federated Median-of-Means UCB (Fed-MOM-UCB) in a federated environment [10].

All the progress shows that MAB methods are becoming increasingly adjustable, scalable, and practical. It gives the paper rich ground to explore UCB, Thompson Sampling, and Bayesian LinUCB for movie recommendation.

DOI: 10.54254/2753-8818/2026.CH30042

3. Methodology

The research methodology is designed into three interconnected phases. The first phase focuses on data processing. This includes loading the MovieLens dataset, creating 18 arms based on movie groups, and constructing feature vectors for the Bayesian Linear UCB algorithm. The second phase is dedicated to model construction, implementing three multi-armed bandit algorithms: the UCB algorithm, the Thompson Sampling algorithm, and the Bayesian Linear UCB algorithm. The third phase involves performance evaluation, comprehensively assessing the algorithms using three metrics: cumulative regret, regret rate, and optimal arm selection frequency.

3.1. Data processing

This research uses the MovieLens 1M dataset for experiments, which includes three core files: user information, movie ratings, and movie metadata. The data preprocessing process is as follows.

For data loading and integration, the research loads three files: `users.dat` (user attributes), `ratings.dat` (rating records), and `movies.dat` (movie information) - Merge rating data with user attribute data through user ID (`user_id`), then merge with movie information through movie ID (`movie_id`) to form a complete dataset containing user-movie-rating-genre - Perform data integrity checks to ensure all required files exist and prevent errors in subsequent processes. For the Arm Definition, the research divides the data into arms based on movie genres. Calculate the average rating of each genre as the true reward value for subsequent regret calculation.

For feature engineering, the research constructs a 12-dimensional context feature vector for each sample, including: - Gender features: 1-dimensional binary encoding (1 for Female, 0 for Male) - Occupation features: 4-dimensional classification encoding (Student/Education, Technical/Professional, Service/Sales, Management/Administration) - Age features: 5-dimensional one-hot encoding (≤18, 19-25, 26-35, 36-45, >45) - Interaction features: 2-dimensional cross features (underage female, young adult male).

For data preparation, the research creates a list of (feature vector, rating) tuples for each arm for easy algorithm invocation - Implement a data cycle index mechanism with shuffling to avoid sequence bias, supporting continuous access to each arm's data during experiments This preprocessing process converts raw recommendation data into an input format that conforms to the contextual multi-armed bandit problem framework, which not only retains key feature information of users and items but also ensures stable operation of the algorithm through standardized processing.

3.2. Model principles and optimization

This study employs the Bayesian LinUCB algorithm as the core model for solving the contextual multi-armed bandit problem in personalized recommendation scenarios.

For hyperparameter optimization, this model uses a grid search approach over a predefined set of values for key parameters, including the prior regularization parameter, the exploration coefficient, and the feature dimension. Then, the dataset is divided into 5 groups, where four groups are used for training and one for validation. This search uses 3 evaluation metrics to evaluate the algorithm’s performance.

The Bayesian LinUCB algorithm maintains a posterior distribution over the linear model weights for each arm. The weights follow a normal distribution, and the mean vector and covariance matrix are updated iteratively (as shown in Formula (1) and (2). When selecting an arm, the algorithm

DOI: 10.54254/2753-8818/2026.CH30042

computes the UCB value for each arm. The UCB value is the sum of the posterior mean prediction and a term proportional to the posterior standard deviation. This UCB value helps the model select toward arms that either have great performance or have high uncertainty.

Update phase, after seeing a reward from selected arms, the parameter of the posterior distribution will be updated by using the conjugate prior property from Bayesian Linear Regression: The inverse covariance matrix is updated by adding the outer product of the context feature vector. The mean vector is updated to include the new observation while keeping in mind the prior belief. With this update rule, the model will keep getting better at guessing the weights of an arm as it learns more and more over time, making its decisions better with each iteration.

$$\sum_{n}^{-1} = \sum_{n-k}^{-1} + \sum_{i=1}^{n} x_i x_i^t + \gamma \cdot I \tag{1}$$

Formula 1 is to update the posterior covariance matrix. The terms on the Right-Hand Side (RHS) are the inverse of the posterior covariance matrix before this batch. This retains the prior information with respect to the parameters. The paper adds all of the sums of the external products with the context vectors of the current batch. It gives the newest data. Finally, the paper adds a regularization term so the covariance matrix will not be singular.

$$\mu_n = \sum_n ((\sum_{n}^{-1} - \lambda \cdot I) \cdot \mu_{n-k} + \sum_{i=1}^{k} r_i x_i \tag{2}$$

Formula 2 shows how to update the posterior mean vector, which is calculated using the updated covariance matrix from the previous step. The terms involved include: the product of the updated inverse covariance matrix (minus a prior precision term) and the prior mean vector, combined with the sum of the products of each reward and its corresponding context vector from the current batch. This operation integrates the new reward-related information with the prior beliefs to refine the mean estimate.

3.3. Evaluation metrics

The final performances of the models are assessed using three metrics: cumulative regret, regret rate, and optimal arm selection frequency.

Cumulative regret for a whole run of an algorithm is the total loss of choosing optimal arms rather than the optimal one throughout the entire length of time. It is computed as the sum over all steps of the difference between the reward of the best arm and the actual chosen arm. This is a direct measure of how much reward the algorithm has given up on. Especially, we'd like to know how much the algorithm spends on exploration and exploitation in total over time.

$$R(T) = \sum_{t=1}^{T} (\mu^* - \mu_{a_t}) \tag{3}$$

Where T is the total number of time steps, r* is the true value of the best arm, and μ* is the true value of the arm a_{t} chosen at step t.

DOI: 10.54254/2753-8818/2026.CH30042

Regret rate is the rate at which the sum of regrets is increasing by step number. It uses a power-law relationship formula 4 to fit the cumulative regret curve. If the regret rate is below 1, it implies that the regret is developing slower than linearly with respect to time, which is advantageous for a Multi-Armed Bandit algorithm. This metric would be necessary if the paper needs long-term behavior and scaling of the application of this algorithm. To get it, the search takes a log-log transformation and linear regression. Step 1: Take logs of steps and the corresponding regret, then fit a linear model to this data, with the slope of the fit being the regret rate α.

$$R(T) \approx C \cdot T^{\alpha} \tag{4}$$

The paper uses R(T) to denote the regret up until time step T, and α is the regret rate. Optimal arm selection frequency looks into how frequently the algorithm chooses the arm that gives the greatest reward (true reward). This is found by dividing the number of times the arm with the most reward was selected by the total number of times an arm is chosen and noted periodically through the experiment. This is going to be a very important metric for what the paper wants to look at, in terms of how fast and reliably the algorithm can get onto the best arm and be on it, and remain on it.

$$\mathrm{F(T)} = \frac{\mathrm{N^*}}{\mathrm{T}} \tag{5}$$

$\mathrm{N^*}$ is the number of times the optimal arm is selected in the first T steps.

4. Experiment and results

4.1. Experimental setup and dataset

The experiments were conducted on a system equipped with an Intel 13th Generation Core i7-13700HX 16-core CPU, 16GB RAM, and NVIDIA GeForce RTX 5060 Laptop Graphics Processing Unit (GPU), running Windows 11. The software environment included Python 3.10 with key libraries: NumPy 2.2.6, pandas 2.3.1, matplotlib 3.10.5, and tqdm 4.67.1.

The study utilizes the MovieLens 1M dataset, consisting of three main files: users.dat (user attributes including gender, age, occupation, and zip code), ratings.dat (user-movie rating records with timestamps), and movies.dat (movie titles and genres). The experimental framework merges ratings.dat and users.dat based on user ID to create a comprehensive dataset containing user attributes and movie ratings. Rather than traditional train-test splits, the study employs an online bandit setting where all available data creates movie-based arms and corresponding feature vectors, with algorithms interacting with data sequentially during experiments. The dataset is processed to define 18 arms based on movie groups, with each arm's average rating serving as the true reward value for subsequent regret calculations.

4.2. Results

The experimental results of Bayesian LinUCB, UCB, and Thompson Sampling on the MovieLens dataset in terms of cumulative regret, regret rate, and optimal arm selection frequency.

DOI: 10.54254/2753-8818/2026.CH30042

Cumulative Regret. In the multi-armed bandit recommendation experiment conducted on the MovieLens dataset, the analysis results for the core evaluation metric Cumulative Regret, as shown in Fig. 1, reveal significant performance differences among the three algorithms. Specifically, the UCB algorithm achieves the optimal performance, with a final cumulative regret value of 817.93; the Thompson Sampling algorithm ranks second, reaching a cumulative regret value of 2776.36; while the Bayesian LinUCB algorithm performs the worst, with a cumulative regret value as high as 34105.02. A further analysis of the dynamic variation trend of cumulative regret shows that the cumulative regret curves of both the UCB algorithm and the Thompson Sampling algorithm exhibit the characteristic of sublinear growth—a trend consistent with the ideal performance of multi-armed bandit algorithms. In sharp contrast, the cumulative regret curve of the Bayesian LinUCB algorithm exhibits a distinct characteristic of linear growth. This implies that, in the experimental scenario of the current MovieLens dataset, the total reward loss of this algorithm increases at a constant rate with the increase in the number of decisions, and its ability to balance exploration and exploitation of optimal actions is relatively weak.

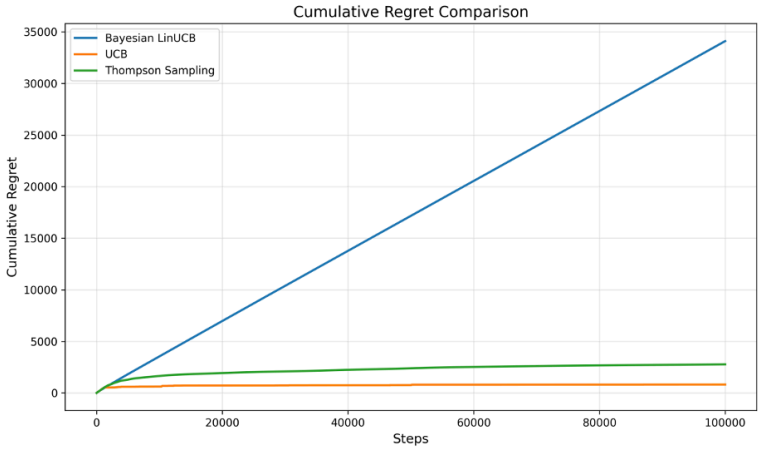

Figure 1. Results of cumulative regret comparison (photo credit: original)

Regret Rate. In the experiment, the experimental statistical results for the key metric regret rate further confirm the performance gaps among the three algorithms, as shown in Fig. 2. Specifically, the UCB algorithm still demonstrates the optimal performance, with a regret rate as low as 0.111. The Thompson Sampling algorithm performs second, with a regret rate of 0.2918. Although higher than that of the UCB algorithm, it remains at a relatively low level.

In contrast, the regret rate of the Bayesian LinUCB algorithm is significantly higher, approximately 1. This value means that the growth rate of its reward loss with the number of decision steps is basically synchronized with the increase in steps, that is, it shows an obvious linear growth trend. More importantly, within the limited number of decision steps set in the experiment, the Bayesian LinUCB algorithm failed to exhibit the sublinear growth characteristic that multi-armed bandit algorithms should possess. Generally speaking, the sublinear growth characteristic is the core embodiment of an algorithm's ability to gradually optimize decisions and reduce the loss growth rate through continuous exploration. However, the performance of this algorithm within this range further indicates that in the scenario of the current dataset, its ability to balance exploration and exploitation is insufficient, making it difficult to effectively slow down the loss growth rate through decision optimization within a limited number of steps.

DOI: 10.54254/2753-8818/2026.CH30042

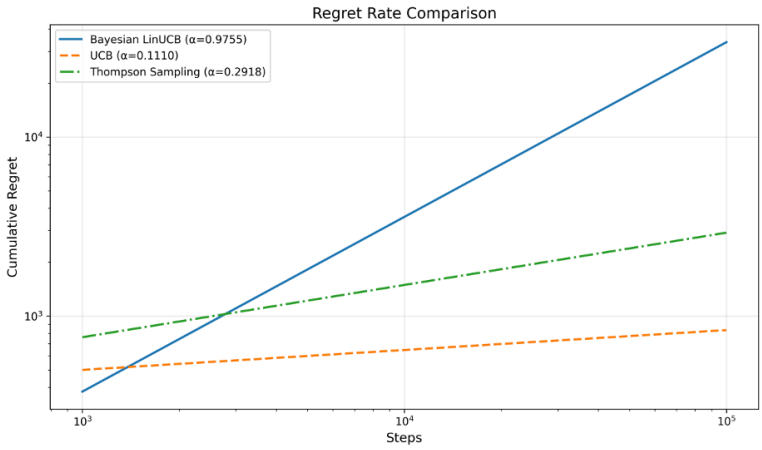

Figure 2. Results of regret rate comparison (photo credit: original)

Optimal Arm Selection Frequency. The statistical results for the key evaluation metric Optimal Arm Selection Frequency show that the three algorithms exhibit a distinct hierarchy in their ability to identify and select optimal actions. As shown in Fig. 3, the UCB algorithm still maintains the optimal performance, with an optimal arm selection frequency as high as 0.9822. The Thompson Sampling algorithm performs second, with an optimal arm selection frequency of 0.9239. Although slightly lower than that of the UCB algorithm, it still maintains high decision accuracy. In contrast, the optimal arm selection frequency of the Bayesian LinUCB algorithm lags significantly, at only 0.1324. This result confirms that in the experimental scenario of the current dataset, the algorithm's exploration efficiency and recognition accuracy for optimal actions are both at a low level.

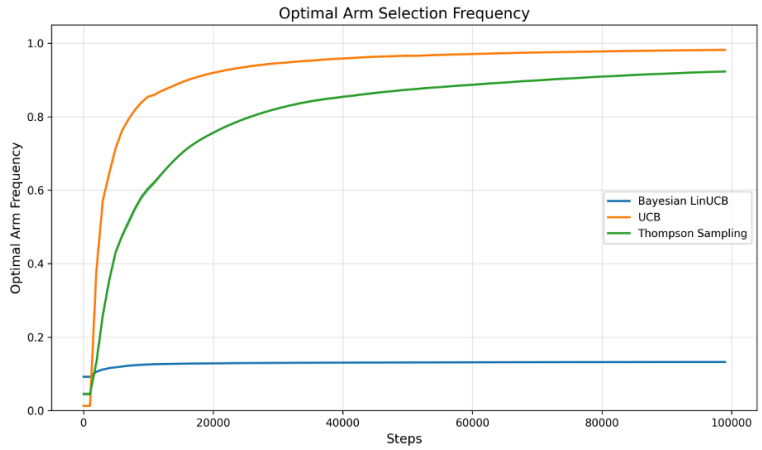

Figure 3. Results of optimal arm selection frequency (photo credit: original)

4.3. Analysis

In the multi-armed bandit recommendation experiment on the MovieLens dataset, a comprehensive analysis of three key metrics—Cumulative Regret, Optimal Arm Selection Frequency, and Regret Rate—clearly defines the performance hierarchy and core differences among the UCB, Thompson Sampling, and Bayesian LinUCB algorithms:

In terms of Cumulative Regret, the UCB algorithm achieves the optimal performance with a final value of 817.93, followed by Thompson Sampling(2776.36), while Bayesian LinUCB(34105.02) performs the worst. In terms of dynamic trends, the cumulative regret curves of UCB and Thompson

DOI: 10.54254/2753-8818/2026.CH30042

Sampling exhibit sublinear growth—a characteristic consistent with the ideal performance of multi-armed bandit algorithms. This indicates that as the number of decision rounds increases, the ability of both algorithms to avoid non-optimal actions continuously improves, and the growth rate of total reward loss gradually slows down. In contrast, Bayesian LinUCB shows linear growth, meaning its reward loss increases at a constant rate with the number of decisions, and its adaptation efficiency to optimal actions is significantly lower.

Regarding Optimal Arm Selection Frequency, the UCB algorithm dominates with a high frequency of 0.9822, demonstrating an extremely strong ability to identify optimal actions. Thompson Sampling maintains a sub-optimal level with a frequency of 0.9239, and its exploration-exploitation balance strategy based on Bayesian inference can stably lock in the optimal arm. In contrast, the 0.1324 frequency of Bayesian LinUCB confirms that its exploration accuracy and decision reliability for optimal actions are at a low level.

From the analysis of the Regret Rate metric, UCB exhibits the optimal loss control ability with a low rate of 0.111, and the growth rate of reward loss is extremely slow. Although Thompson Sampling (0.2918) is slightly higher, it can still control the loss rhythm through strategy balance, and both conform to the core characteristic of sublinear growth. However, the regret rate of Bayesian LinUCB is approximately 1, meaning the growth rate of its reward loss is synchronized with the number of decision steps, showing an obvious linear growth trend. Moreover, within the limited number of steps set in the experiment, Bayesian LinUCB completely fails to exhibit the sublinear growth characteristic that multi-armed bandit algorithms should possess, reflecting the failure of its exploration-exploitation balance mechanism in the current dataset scenario.

5. Conclusion

The comprehensive analysis of three key metrics—cumulative regret, optimal arm selection frequency, and regret rate—establishes a clear performance hierarchy among the UCB, Thompson Sampling, and Bayesian LinUCB algorithms in the MovieLens dataset recommendation scenario. UCB achieves optimal performance with the lowest cumulative regret (817.93) and highest optimal arm selection frequency (0.9822), demonstrating superior loss control and optimal action recognition capabilities. Thompson Sampling maintains competitive performance as the second-best algorithm with moderate cumulative regret (2776.36) and selection frequency (0.9239), while both algorithms exhibit the crucial sublinear growth characteristic essential for multi-armed bandit effectiveness. The sublinear growth patterns of UCB and Thompson Sampling create a positive feedback cycle of improved exploration efficiency, reduced loss growth rates, and increased optimal action recognition, confirming their theoretical advantages and practical applicability in recommendation tasks.

In contrast, Bayesian LinUCB demonstrates significantly inferior performance across all evaluation dimensions, recording the highest cumulative regret (34105.02), lowest optimal arm selection frequency (0.1324), and a regret rate approximating 1, indicating problematic linear growth rather than the desired sublinear pattern. This linear growth characteristic suggests fundamental limitations in the algorithm's exploration-exploitation balance mechanism within the current dataset scenario, highlighting the critical importance of algorithm-dataset compatibility in recommendation systems. The findings emphasize that sublinear growth capability directly determines revenue stability and long-term performance in recommendation tasks, while also demonstrating that algorithm adaptability is strongly correlated with specific dataset characteristics and feature distributions.

DOI: 10.54254/2753-8818/2026.CH30042

References

[1] Auer, P., Cesa-Bianchi, N., Fischer, P.: Finite-time analysis of the multiarmed bandit problem. Machine Learning 47(2), 235–256 (2002).
[2] Li, L., Chu, W., Langford, J., Schapire, R.E.: A contextual-bandit approach to personalized news article recommendation. In: Proceedings of the 19th International Conference on World Wide Web, pp. 661–670 (2010).
[3] Agrawal, S., Goyal, N.: Thompson sampling for contextual bandits with linear payoffs. In: International Conference on Machine Learning, pp. 127–135. PMLR (2013).
[4] Wei, L., Srivastava, V.: Nonstationary stochastic multiarmed bandits: UCB policies and minimax regret. arXiv preprint arXiv: 2101.08980 (2021).
[5] Zhu, J., Liu, J.: Distributed multi-armed bandit over arbitrary undirected graphs. In: 2021 60th IEEE Conference on Decision and Control (CDC), pp. 6976–6981. IEEE (2021).
[6] Qiu, S., Wang, L., Bai, C., Yang, Z., Wang, Z.: Contrastive ucb: Provably efficient contrastive self-supervised learning in online reinforcement learning. In: International Conference on Machine Learning, pp. 18168–18210. PMLR (2022).
[7] Zhu, R.J., Qiu, Y.: UCB Exploration for Fixed-Budget Bayesian Best Arm Identification. arXiv preprint arXiv: 2408.04869 (2024).
[8] Elumar, E.C., Tekin, C., Yağan, O.: Multi-armed bandits with costly probes. IEEE Transactions on Information Theory (2024).
[9] Wu, H., Xu, Y., Cao, S., Liu, J., Takakura, H., Norio, S.: Sleeping Multi-Armed Bandit-Based Path Selection in Space-Ground Semantic Communication Networks. In: 2025 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1–6. IEEE (2025).
[10] Saday, A., Demirel, İ., Yıldırım, Y., Tekin, C.: Federated multi-armed bandits under byzantine attacks. IEEE Transactions on Artificial Intelligence (2025).