Proceedings of CONF-MLA 2026 Symposium: Intelligent Systems and Automation: AI Models, IoT, and Robotic Algorithms
DOI: 10.54254/2755-2721/2026.LD39100

# *The Comparison of Multi-Armed-Bandit Algorithms in Movie Recommendation Systems*

**Ruiyang Ye**

*Department of Data Science and Artificial Intelligence, The Hong Kong Polytechnic University, Hong Kong, China*
*ruiyang2026hk@163.com*

***Abstract.*** With the development of information technology, especially the Internet, people are rapidly moving from the era of information scarcity to the era of information overload. Recommendation systems have thus rapidly developed, which aims to find the optimization of the experience for users. The Multi-Armed-Bandit (MAB) algorithms belong to the category of reinforcement learning , always trying to optimize the balance between exploration and exploitation trade-offs. This fundamental principle successfully coincides the application of recommendation systems. By using a large number of real user ratings of movies as the data-set, this paper compares the performances of various MAB algorithms, ε-greedy, Explore-Then-Commit (ETC), Upper Confidence Bound (UCB), Asymptotically optimal UCB and Thompson Sampling (TS) in the movie recommendation system. Through the performance of different algorithms on the same data, it is concluded that the TS algorithm performs relatively optimally in improving the efficiency of the system. Since the basic principle of TS algorithm utilizes Bayesian inference, the balance of each selection trade-off can more probably be better than other algorithms. TS algorithm has a strong flexibility and can perform a better balance when facing different environments. In addition, the methodological framework of this paper can be drawn upon in more complex scenarios and environments.

***Keywords:*** Multi-Armed Bandit (MAB), Movie Recommendation Systems, Thompson Sampling (TS)

## 1. Introduction

Reinforcement learning (RL) is a machine learning method. Reinforcement learning focuses on the situation where an intelligent agent interacts with an environment based on prior experiences, similar to humans [1]. The goal of reinforcement learning is for an intelligent agent to learn a policy that defines what action should be taken in a given state to maximize the cumulative reward.

In the MAB problem, the intelligent agent needs to find a balance between uncertainty and known information. Similarly, in the broader reinforcement learning task, the intelligent agent faces a similar choice. MAB problems typically focus on immediate rewards, while RL often involves more complex delayed reward situations; both aim to maximize long-term rewards [2]. The algorithms in the MAB Problem can be applied to certain specific contexts in reinforcement

© 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: 10.54254/2755-2721/2026.LD39100

learning, especially in those cases where there is immediate feedback and the environment does not change much or the state space is small, such as language processing [3], drug discovery [4], and unbalanced networks [5]. The primary objective of this paper is to conduct a comparison of the efficiency of various MAB algorithms in movie recommendation system.A more comprehensive review of the literature on recommendation systems is been examined, especially the articles on algorithm performance comparison.

The paper compares several commonly popular MAB algorithms. Also the effect of parameters on the performance of the algorithms is further analyzed by changing the hyper-parameters of several algorithms. In the concluding phase of the experiments, conclusions regarding each algorithm will be drawn, providing promising suggestions for optimizing the efficiency of recommendation systems.

## 2. Literature review

### 2.1. Multi-Armed Bandit problems

The Multi-Armed Bandit (MAB) problem is concerned with sequential decision making where an algorithm chooses actions (arms) and receives rewards from the environment [6]. Performance of algorithms depends on the reward structure (IID, adversarial, constrained adversarial, random-process) [7]. Regret is one of the key ideas, which measures the reward loss due to the selection of suboptimal arms compared to always choosing the optimal arm [8]. The goal of MAB research is therefore to minimize the cumulative regret, ideally sublinearly in the number of rounds [9]. In this study, we consider stochastic bandits with real-world movie ratings as reward data. There is a trade-off between exploration (the process of gathering information on different arms) and exploitation (the process of choosing the arm with the highest estimated reward) [10, 11].

### 2.2. MAB algorithms in recommendation system

MAB algorithms have been used in online recommendation, personalized content selection, A/B testing, content ranking and dynamic pricing. Research shows that these algorithms can improve the recommendation efficiency by learning from the user feedback and adapting the selection strategies [12, 13]. Different algorithms have different ways of balancing exploration and exploitation . The appropriateness of a given algorithm thus depends on the application context

### 2.3. Gaps in current research

MAB algorithms are applied to movie recommendation and the recommendation performance is reported to be improved in previous studies [14]. Extensions of the Beta distribution such as Inheritance Thompson Sampling have explored variants in the Beta distribution to reduce cumulative loss [15]. Other work has studied the combination of MAB with reinforcement learning [16], developed models for optimizing positional rewards [17], and applied bandit methods to advertising recommendations [18]. However, despite these advances, further comparisons are needed to evaluate the relative performance of different algorithms in specific recommendation settings. Therefore, this study investigates five representative MAB algorithms by utilizing the movie rating data, and cumulative regret to find effective strategies to improve the recommendation performance and personalization.

DOI: 10.54254/2755-2721/2026.LD39100

## 3. Experiment principle

### 3.1. ε-greedy algorithm

The ε-greedy algorithm is a trade-off between exploration and exploitation with a hyperparameter epsilon [19]. At each round, the algorithm picks a random arm with probability $\epsilon$ and the arm with highest estimated reward with probability $1 - \epsilon$. A randomly generated number decides which action is taken.

### 3.2. Explore-Then-Commit algorithm

The Explore-Then-Commit (ETC) algorithm has two phases: In exploration, each arm is pulled $m$ times, which gives $mk$ exploration rounds for a $k$-arm bandit. It then selects the arm with the largest empirical mean reward and repeatedly pulls it during the commit phase..

### 3.3. UCB-based algorithm

Upper Confidence Bound (UCB) Algorithm calculates an upper confidence bound for each action, to decide the next action to choose.The result of calculation is called index, which is defined as:

$$UCB_i(m-1) = \hat{u}_i(m-1) + \frac{B}{2}\sqrt{\frac{\tau logn}{N_i(m-1)}} \tag{1}$$

Where B is the discrepancy between the largest practical reward value and the minimal practical reward value, $\hat{u}_i(m-1)$ is the average reward of arm i when it is t-1 round, $N_i(m-1)$ is the number of selecting arm i when t-1 rounds finished, τ is a hyper-parameter determining the realm of upper confidence bound.

Initially each arm would be selected once, to avoid the situation that the time T of a certain arm could be zero. Algorithm then chooses the arm with the highest index value, updating the reward of each round, and repeating these steps til the total rounds finished. For this algorithm, the index (same definition to UCB) adjusts to:

$$UCB_i(m-1) = \hat{u}_i(m-1) + \frac{B}{2}\sqrt{\frac{\tau log(f(m))}{N_i(m-1)}} \tag{2}$$

Where $f(m) = 1 + m(\log(m))^2$. The meanings of other parameters are the same as in equation (1).

### 3.4. Thompson Sampling algorithm

Thompson Sampling (TS) is a Bayesian inference based approach that balances exploration and exploitation. The algorithm maintains a prior distribution Beta for every arm and samples a reward estimate from the posterior distribution. It chooses the arm with the largest sampled value and updates the distribution for that arm based on the observed outcome. These steps are repeated until all experimental rounds are completed, as shown in Equations (3) and (4).

$$B(\alpha_i, \beta_i) = \int_0^1 t^{\alpha_i - 1}(1-t)^{\beta_i - 1} dt \tag{3}$$

DOI: 10.54254/2755-2721/2026.LD39100

Where $\alpha_i$ is the time when success occurs, and βi is the time when failure occurs. Algorithm then samples a value $\theta_i$ from all arms with Beta distribution, choosing the arm with the highest value $\theta_i$. Now updates parameters:

$$\alpha_i = \alpha_i + 1, \beta_i = \beta_i, \text{if success (obtains reward)}$$

$$\beta_i = \beta_i + 1,\ \alpha_i\ =\ \alpha_i,\ \text{if failure (obtains no reward)} \tag{4}$$

Repeat these steps til the total rounds finished.

## 4. Experimental methodology

### 4.1. System introduction

Movie recommendation system is an important application of machine learning technology in the entertainment industry, and its goal is to automatically recommend movies that users may be interested in based on their personal preferences and behavioral data. Such a system not only enhances user experience, but also increases user stickiness and the commercial value of the platform. The main challenge facing movie recommendation systems is how to find the content that best matches the user's preferences in a massive movie library. Machine learning technology provides an automated solution that can generate accurate recommendations based on a large amount of user data.

### 4.2. Experimental framework & data sources

In this movie recommendation system, the author will use each movie's genre as an arm in a Multi-armed Bandit problem, and each movie's rating as a reward corresponding to each arm. Five different algorithms mentioned above will be used in this system, which will be used to perform a comparison of the degree of superiority of the algorithms. The number of experimental rounds for each algorithm may vary. The comparison of algorithms will include the comparison of one algorithm with different parameters, as well as the comparison of different algorithms with the same parameters.

There are a total of 18 movie genres, namely Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War and Western.Since a movie may have multiple genres, thus making it impossible to count the rewards corresponding to the arm, the data should be self-processed in some way to separate the multiple genres of a particular movie.

Movie ratings are real ratings from various users of MovieLens, ranging from 1 to 5. These files contain 1,000,209 anonymous ratings of approximately 3,900 movies made by 6,040 MovieLens users who joined MovieLens in 2000 [20]. With 3 being the median, meaning that users are more likely to click on movies with ratings higher than 3, and are not interested in movies with ratings lower than 3, and thus will not click on them.

DOI: 10.54254/2755-2721/2026.LD39100

## 4.3. Experimental results

### 4.3.1. ε-greedy algorithm

ε is the hyper-parameter of the algorithm that controls the probability of choosing an arm at random, and for better parameter comparisons, ε is set to five different values:0.1, 0.2, 0.3, 0.4, 0.5, and the average cumulative regrets are output as in Figure 1.

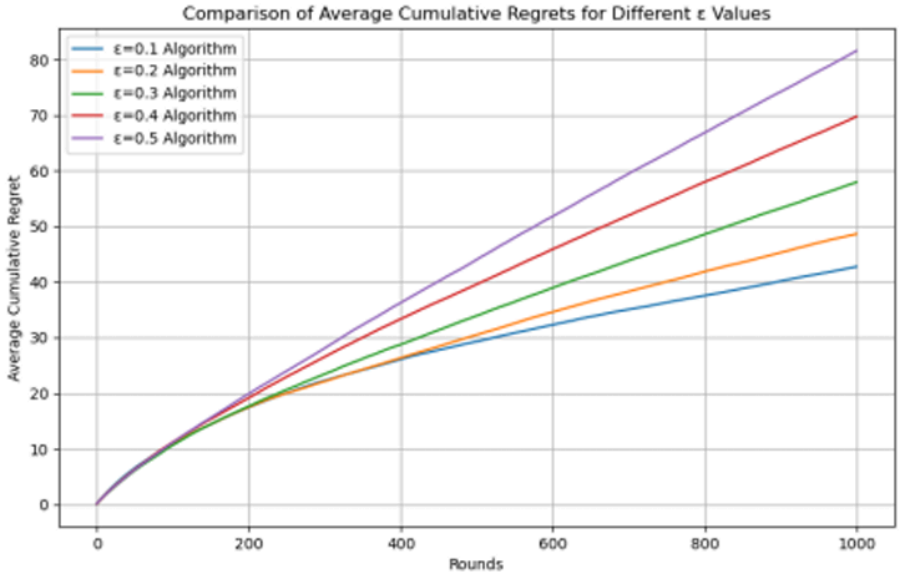



Figure 1. The figure shows the comparison of average cumulative regrets for different values

### 4.3.2. Explore-Then-Commit algorithm

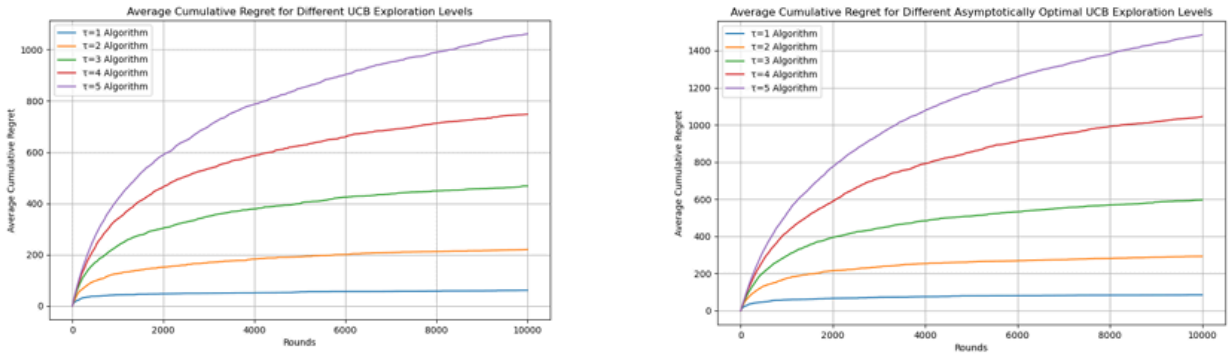
The exploration phase of the ETC algorithm is one of the key parts of the algorithm's success because it determines how well the algorithm explores each arm. The goal of the Explore phase is to gather enough information to evaluate which arm might be the optimal.

Choosing the number of explorations for each arm requires consideration of the total number of rounds the algorithm will run, and the number of explorations should be a reasonable proportion of the total number of rounds to ensure that there is enough data to evaluate each arm.An exploration phase that is too short may result in an inaccurate assessment of the arm, leading to the selection of a sub-optimal arm in the utilization phase. An exploration phase that is too long may reduce the number of rounds in the utilization phase and affect the overall performance. Ten experiments were done, each experiment was conducted in rounds of 10,0000.

### 4.3.3. Upper confidence bound algorithm & Asymptotically Optimal UCB algorithm

Figure 2 compares average cumulative regret of UCB and Asymptotically Optimal UCB algorithms over different values of exploration parameter. The results show the influence of the level of exploration on performance of the algorithms and enable to compare the sensitivity of the two algorithms on change of parameters.

DOI: 10.54254/2755-2721/2026.LD39100




(a)The comparison of average cumulative regrets for different exploration levels

(b)The comparison of average cumulative regrets for different exploration levels

Figure 2. Comparison of UCB and Asymptotically Optimal UCB under different exploration parameters

### 4.3.4. Thompson Sampling algorithm

The TS algorithm uses the Beta distribution as the prior distribution for the binary reward case, and the speed of convergence of the TS algorithm depends on the characteristics of the problem and the choice of the prior distribution. In some cases, the algorithm may take a longer time to converge to the optimal policy. In some dynamic environments, the optimal policy may change.The TS algorithm needs to be able to adapt to these changes, and therefore the algorithm parameters need to be continuously monitored and adjusted.

This study compares the TS algorithms by varying the total number of rounds of experiments, 1000, 10000, 100000 respectively. The curves of average cumulative regret may change with different experimental rounds (i.e. at what value is it logarithmically distributed). The results are displayed in Figure 3.

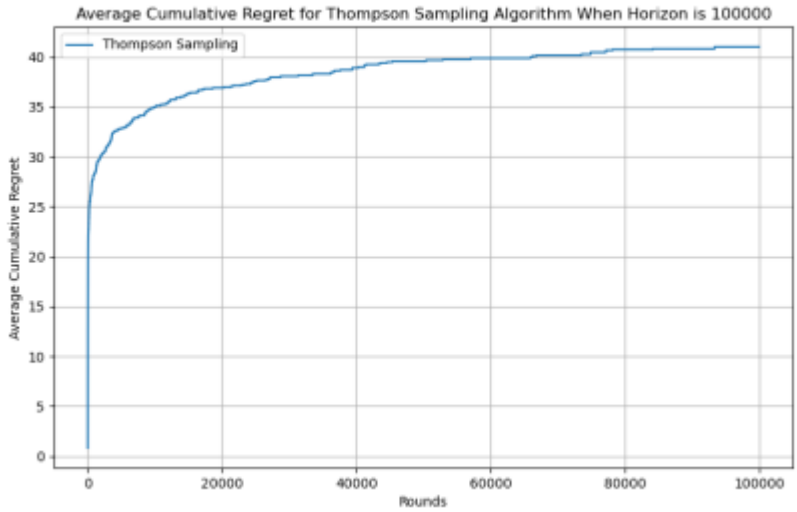



Figure 3. The figure shows the average cumulative regret when total rounds are 100,000

DOI: 10.54254/2755-2721/2026.LD39100

### 4.3.5. Algorithms comparison

For the comparison algorithm experiments, the number of experiments for each algorithm is 10 and the number of experimental rounds is 1000. for the ε - greedy Algorithm, the hyper-parameter ε is set to 0.1. for the ETC Algorithm, the number of explorations for each arm in the exploration phase was set as (N is the value of total rounds). For UCB and Asymptotically Optimal UCB , the exploration degree c is set to 2. The result is shown as Figure 4.

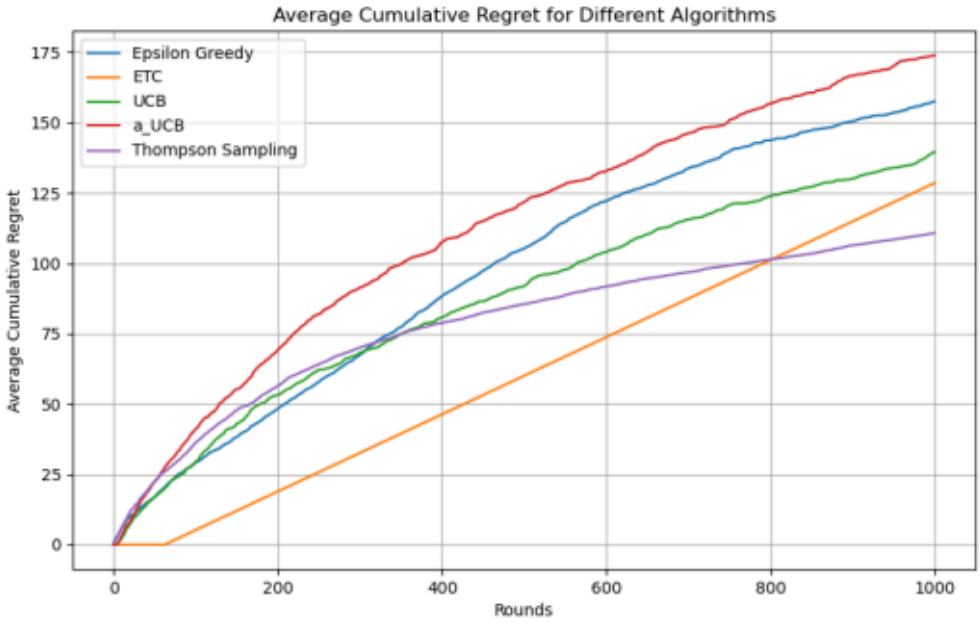



Figure 4. The figure shows the comparison of average cumulative regret for different algorithms

## 5. Comparative analysis

By comparing the results of the five algorithms in one figure, it can be seen that the TS algorithm performs the best over a long period of time. the ETC algorithm rises linearly and eventually its average cumulative regret exceeds that of the other algorithms. ε-greedy algorithm initially performs better than the UCB algorithm, but as the number of rounds increases the UCB algorithm increases more consistently and thus outperforms ε-greedy algorithm. The Asymptotically Optimal UCB Algorithm presents the highest average cumulative regret over 1000 rounds, which shows that it is not superior under the movie rating recommendation system. The TS algorithm is the optimal algorithm in the long run. It has the lowest degree of regret growth and will eventually outperform the other four algorithms and level off, indicating that the TS algorithm is a relatively optimal algorithm in this system.The drawbacks of the ETC algorithm are mainly due to its non-adaptive nature and fixed exploration phase. ETC performs an equal amount of exploration for each arm in the initial phase, causing the situation that the best arm may be prematurely excluded if it performs poorly in the exploration phase. In contrast, UCB makes a judgment based on its upper confidence bound at each selection, while TS is based on Bayesian inference with a probability distribution for each arm. Therefore, these two algorithms are able to dynamically adjust the balance between exploration and exploitation, and usually outperform the ETC algorithm in terms of long-term performance.

DOI: 10.54254/2755-2721/2026.LD39100

## 6. Conclusions

It benchmark five MAB algorithms on a real-world dataset of movie ratings. The results indicate that TS has the best overall performance, balancing exploration and exploitation and adapting well to changing environments. However, the cumulative regret of ETC is linear and not suitable for movie recommendation. The results suggest that TS has the potential to be applied in personalized recommendation, online learning, and advertisement placement. There are however several limitations to this study. We only compare five widely accepted MAB algorithms, so there is room for comparison with other methods.

The promise of using MAB algorithms to recommend movies remains high. Further research can be on the contextual MAB by including user features and the environmental information and combining deep learning to improve the accuracy of the recommendations. New technologies like augmented and virtual reality could also provide more immersive and personalized experiences. With the increasing computation power, new variants of MABs may improve the efficiency and adaptability of recommendations for different application scenarios.

## References

[1] Naeem, M., Rizvi, S. T. H., & Coronato, A. (2020). A gentle introduction to reinforcement learning and its application in different fields. *IEEE Access, 8*, 209320–209344. doi: 10.1109/ACCESS.2020.3038605.
[2] Nishimura, Y. Ad recommender system analysis by the multi-armed bandit problem.
[3] Lee, J. D., Lei, Q., Saunshi, N., & Zhuo, J. (2021). Predicting what you already know helps: Provable self-supervised learning. *Proc. Adv. Neural Inf. Process. Syst., 34*, 309–323.
[4] Ramsundar, B., Kearnes, S., Riley, P., Webster, D., Konerding, D., & Pande, V. (2015). Massively multitask networks for drug discovery.
[5] Odeyomi, O. T. (2023). Online learning of time-varying unbalanced networks in non-convex environments: A multi-armed bandit approach. *IEEE Access, 11*, 15567–15577. doi: 10.1109/ACCESS.2023.3244070.
[6] Lattimore, T., & Szepesvári, C. (2020). *Bandit algorithms*. Cambridge University Press.
[7] Slivkins, A. (2022). *Introduction to multi-armed bandits*.
[8] Wei, L., & Srivastava, V. (2024). Nonstationary stochastic bandits: UCB policies and minimax regret. *IEEE Open Journal of Control Systems, 3*, 128–142. doi: 10.1109/OJCSYS.2024.3372929
[9] Liu, P., Wang, L., & Chen, Y. Sub-linear dynamic regrets for aggregative games with feedback delays and communication delays. *IEEE Transactions on Control of Network Systems*. doi: 10.1109/TCNS.2024.3432250
[10] Chen, W., et al. (2019, January). Structure-based suggestive exploration: A new approach for effective exploration of large networks. *IEEE Transactions on Visualization and Computer Graphics, 25*(1), 555–565. doi: 10.1109/TVCG.2018.2865139
[11] Bommaraveni, S., Gautam, S., Vu, T. X., & Chatzinotas, S. (2021). On the exploration and exploitation trade-off in cooperative caching-enabled networks. *2021 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, Nanjing, China, 1–7. doi: 10.1109/WCNCW49093.2021.9419984.
[12] Yan, C., Han, H., Wang, Z., & Zhang, Y. (2021). Two-phase multi-armed bandit for online recommendation. *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, Porto, Portugal,1–8. doi: 10.1109/DSAA53316.2021.9564225.
[13] Huang, W., Labille, K., Wu, X., Lee, D., & Heffernan, N. (2021). Fairness-aware bandit-based recommendation. *2021 IEEE International Conference on Big Data (Big Data)*, Orlando, FL, USA,1273–1278. doi: 10.1109/BigData52589.2021.9671959.
[14] Jian, Y. (2024). Optimizing movie recommendation systems with multi-armed bandit algorithms. *Highlights in Science, Engineering and Technology, 94*, 175–181.
[15] Shi, J.-x. (2024). Inheriting Thompson sampling for movie recommendation. *Proceedings of the 1st International Conference on Engineering Management, Information Technology and Intelligence*, n. pag.
[16] Silva, N., Werneck, H., Silva, T., Pereira, A. C. M., & Rocha, L. (2022). Multi-armed bandits in recommendation systems: A survey of the state-of-the-art and future directions. *Expert Systems with Applications, 197*, 116669. https://doi.org/10.1016/j.eswa.2022.116669

DOI: 10.54254/2755-2721/2026.LD39100

[17] Ravi, A. N., Poduval, P., & Moharir, S. (2020). Unreliable multi-armed bandits: A novel approach to recommendation systems. *2020 International Conference on COMmunication Systems & NETworkS (COMSNETS)*, Bengaluru, India, 650–653. https://doi.org/10.1109/COMSNETS48256.2020.9027470

[18] Lacerda, A., Veloso, A., & Ziviani, N. (2015). Adding value to daily-deals recommendation: Multi-armed bandits to match customers and deals. *2015 Brazilian Conference on Intelligent Systems (BRACIS)*, Natal, Brazil, 216–221. https://doi.org/10.1109/BRACIS.2015.66

[19] Nguyen, Q., Teku, N., & Bose, T. (2021). Epsilon greedy strategy for hyper parameters tuning of a neural network equalizer. *2021 12th International Symposium on Image and Signal Processing and Analysis (ISPA)*, Zagreb, Croatia, 209–212. https://doi.org/10.1109/ISPA52656.2021.9552055

[20] Harper, F. M., & Konstan, J. A. (2015). The MovieLens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS), 5*(4), Article 19. https://doi.org/10.1145/2827872