Proceedings of the 4th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/54/20241140

# Tackling the cold start issue in movie recommendations with a refined epsilon-greedy approach

**Enqi Ouyang**

Institute of Future Tech, South China University of Technology, Guangzhou, 511442, China

202130310664@mail.scut.edu.cn

**Abstract.** With the rapid growth of the Internet and the consequent surge in data, the current era is characterized by information overload. As the domain of data processing and storage expands, recommendation systems have become pivotal tools in navigating this deluge, assisting users in filtering through vast information landscapes. A notable segment of this is movie recommendation systems. As living standards rise, so does the demand for cinematic experiences. Enhancing and refining the methodologies of these recommendation systems is, therefore, of significant value. However, a consistent challenge is the 'cold start' problem encountered when new users join. Without prior viewing records or preferences, these users pose a dilemma for the system: how to offer relevant recommendations without historical data? Addressing this challenge, this paper proposes a unique method grounded in the N-armed bandit model, introducing an enhanced Epsilon-greedy algorithm specifically designed for movie recommendations for such users. By adjusting dynamically based on real-time user feedback, the algorithm aims to continuously hone its recommendation quality, ensuring a consistently better user experience.

**Keywords:** Epsilon-Greedy Algorithm, Cold-Start Problem, Movie Recommendation, Multi-Armed Bandits.

## 1. Introduction

In today's digital age, the rapid growth of the Internet is accompanied by an unprecedented surge in data. With advancements in data processing techniques and increased storage capacities, we're witnessing an escalating issue of information overload. Recommendation systems have emerged as a crucial tool to mitigate this challenge. Over the years, these systems have been fine-tuned to address a myriad of data-related problems, enabling swift retrieval of related products and web pages to suggest them to users.

A notable subset of this technology focuses on movie recommendations. With the enhancement in the quality of life, the appetite for cinematic experiences has amplified. Statistics from 2016 show that the number of online movie-goers skyrocketed to 1.058 billion [1]. For consumers, recommendation systems curate films tailored to their interests, and for businesses, they serve as a lever to boost profitability. Thus, refining the strategies of movie recommendation systems represents a valuable avenue of technological research. Yet, a stumbling block exists: the cold start problem for new users [2]. When users with no prior viewing records join the platform, the system grapples with making effective

DOI: 10.54254/2755-2721/54/20241140

suggestions. Traditional content-based recommendation approaches, such as collaborative filtering algorithms, often fall short in these scenarios [3].

A straightforward solution has been to resort to random suggestions. This paper, however, introduces an N-arm slot machine model and presents an enhanced epsilon-greedy algorithm. This method recommends movies to new platform users based on comprehensive data, dynamically adjusting its strategy with shifts in the platform's aggregate data, thereby continually honing the algorithm's recommendation precision.

## 2. Theoretical Background and Algorithmic Advancements

### *2.1. Multi-armed Bandits*

The prototype of this problem is a multi-arm slot machine, which has $k$ swing arms. After the player invests a game coin, he can shake any rocker arm. Each rocker arm returns a certain amount of game coin with a certain probability as a return, and the winning probability of each rocker arm is different and may change. The player's goal is to maximize the cumulative return within $n$ rounds through certain strategies.

Here, set in each round t, the player chooses the rocker arm is $A_t$, and the return is $X_t$, using cumulative regret $R_n$ to measure the effect of policy π, which is the difference between the total expected reward using policy π for n rounds and the total expected reward collected by the learner over n rounds and the goal is to get the minimum $R_n$. The regret relative to a set of policies $\Pi$ is the maximum regret relative to any policy $\pi \in \Pi$ in the set. Let $\hat{\mu}_i(t)$ be the average reward received from arm $i$ after round $t$, which is written formally as [4].

$$\hat{\mu}_i(t) = \frac{1}{T_i(t)} \sum_{s=1}^{t} \mathbb{I}\{A_s = i\} X_s \tag{1}$$

where $T_i(t) = \sum_{s=1}^{t} \mathbb{I}\{A_s = i\}$ is the number of times action $i$ has been played after round $t$. Then let $\mu^*(t) = \max_i \mu_i(t)$ be the largest mean of all arms.

The regret over n rounds becomes.

$$R_n = n\,\mu^*(t) - E\left[\sum_{t=1}^{n} X_t\right] \tag{2}$$

Or.

$$R_n = \sum_i \Delta_i E[T_i(n)] \tag{3}$$

where $\Delta_i(t) = \mu^*(t) - \hat{\mu}_i(t)$ is the suboptimality gap or action gap or immediate regret of action $i$.

Therefore, the specific description of multi-armed bandits in this paper are as follows:

**Input:** The number of arm $k$
The function of reward $r$
The number of horizon n

**Process:**

**Begin**
$Regret_n = 0$ ;
$reward_n = 0$;
$\forall i = 1, 2, \ldots, k: T_i = 0,\ \hat{\mu}_i = 0$ ;
**for** t =1, 2, …, n **do**
Choose an arm $A_t$
Get the reward $X_t$
Update the counter $T_i$ and the average reward $\hat{\mu}_i$
Update the cumulative reward $reward_n$
Update the cumulative regret $Regret_n$

DOI: 10.54254/2755-2721/54/20241140

10: **end for**
**Output:** The cumulative regret $Regret_n$
The cumulative reward $reward_n$

### *2.2. Epsilon-Greedy Algorithm: A Traditional Approach*

In the algorithm process described above, how to select the arms at time $t$ is still not mentioned, then a strategy is needed to select the arms at each moment, which involves the classic problem in the multi-arm bandit -- the balance between exploration and exploitation, which needs to be considered in the design of the policy.

Exploration refers to the use of certain methods to achieve the exploration conditions: randomly or sequentially select an arm to obtain more data to provide data support for the utilization stage [5]. Exploitation refers to selecting the arm with the most average reward under known conditions to improve the accuracy of the policy when the exploitation condition is reached.

Therefore, it is necessary to use the algorithm to balance the number of exploration and exploitation in the multi-arm bandit, so as to maximize the cumulative reward and minimize the cumulative regret.

The epsilon-greedy algorithm is a branch of the greedy algorithm. If fully greedy is used, there is only the exploitation stage, that is, an arm with most average reward is selected at every moment without exploration [6]. The Epsilon-greedy algorithm adds some noise to the original greedy algorithm, that is, an arm is randomly selected for exploration with a certain probability.

A quantitative epsilon is defined in this algorithm to determine whether the moment is to be explored or exploited [7]. At every moment, a number between 0 and 1 will be randomly generated. When this number is greater than epsilon, it is the exploration stage, that is, an arm will be randomly selected; when this number is less than epsilon, it is the exploitation stage, that is, the arm with the largest current average reward will be selected. The specific description of this algorithm is as follows:

**Input:** The number of arm $\boldsymbol{k}$
The function of reward $\boldsymbol{r}$
The number of horizon **n**
The value of the decay rate $\boldsymbol{c}$

**Process:**
**Begin**
$Regret_n = 0$;
$reward_n = 0$;
$\forall i = 1, 2, \ldots, k: T_i = 0,\ \hat{\mu}_i = 0$ ;
**for** t =1, 2, …, n **do**
**If** rand() < $\epsilon$ then
$A_t =$ from 1, 2, …, $k$ is randomly selected with uniform distribution
**else**
$A_t = \arg max_i\ \hat{\mu}_i(t)$
**end if**
$T_{A_t}$ += 1;
$\hat{\mu}_{A_t} = \frac{\hat{\mu}_{A_t} \times (T_{A_t} - 1) + r(t)}{T_{A_t}}$ ;
$reward_n$+= $\hat{\mu}_{A_t}$
$Regret_n = Regret_n + \max \hat{\mu}_i - \hat{\mu}_{A_t}$
**end for**
**Output:** The cumulative regret $Regret_n$
The cumulative reward $reward_n$

### *2.3. Refinements to the Epsilon-Greedy Algorithm: Epsilon-decreasing greedy*

In this paper, an improved Epsilon-greedy algorithm is proposed, which is a decreasing epsilon-greedy algorithm, and uses the inverse function of time to decrease the value of epsilon to make regret converge, so as to optimize the strategy.

DOI: 10.54254/2755-2721/54/20241140

In the traditional Epsilon-greedy strategy, the value of epsilon is fixed. Moreover, it can be seen from the experimental results in Figure 1 and Figure 2 that if the value of epsilon is too large, it will promote convergence but lead to excessive exploration, resulting in poor recommendation effect; however, if the value of epsilon is too small, it will lead to insufficient exploration. The convergence speed of the algorithm is too slow [8]. And the fixed epsilon does not make regret converge, but increases linearly.

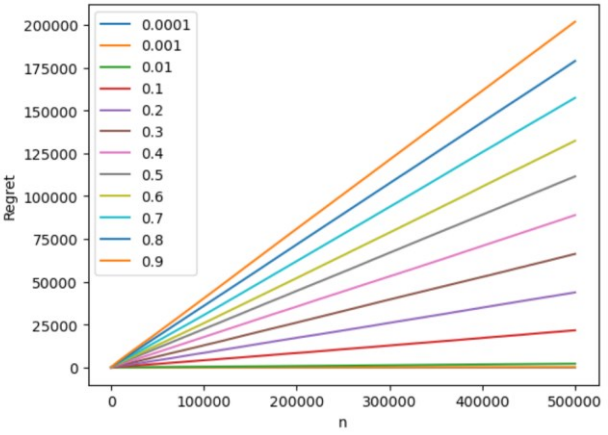


**Figure 1.** Cumulative regret values for different epsilon (Photo/Picture credit: Original).

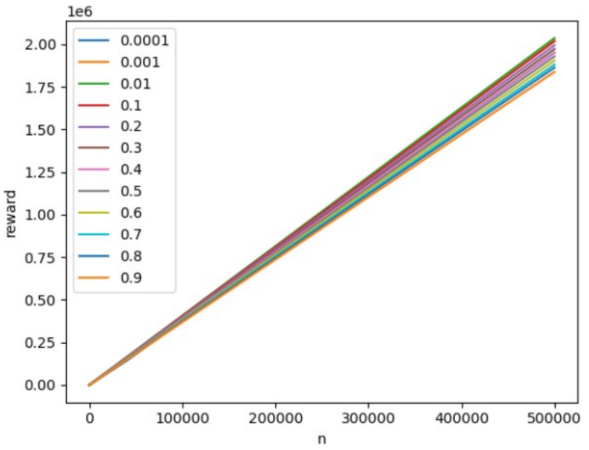


**Figure 2.** Cumulative reward values for different epsilon (Photo/Picture credit: Original).

In the improved Epsilon-greedy algorithm, taking the inverse function of time to decrease epsilon makes epsilon change from large to small, and the selection of the algorithm changes gradually from the initial high exploration to the later high exploitation, so as to balance exploration and exploitation. The main goal of this strategy is to explore more in the initial phase, then gradually reduce the exploration over time, and increase the exploitation of known best moves [9].

The formula for calculating the ε value of the inverse function decline strategy is usually as follows.

$$\varepsilon(t) = \frac{\varepsilon(0)}{[c \times (t+1)]} \tag{4}$$

The epsilon-greedy algorithm is improved as follows:

**Input:** The number of arm $\boldsymbol{k}$
The function of reward $\boldsymbol{r}$
The number of horizon **n**
The value of the decay rate $\boldsymbol{c}$

**Process:**

DOI: 10.54254/2755-2721/54/20241140

**Begin**
$Regret_n = 0$;
$reward_n = 0$;
$\forall i = 1, 2, \ldots, k: T_i = 0,\ \hat{\mu}_i = 0$ ;
**for** t =1, 2, …, n **do**
$\epsilon = 1/(1 + ct)$
**If** rand() < $\epsilon$ then
$A_t =$ from 1, 2, …, $k$ is randomly selected with uniform distribution
**else**
$A_t = \arg max_i\ \hat{\mu}_i(t)$
**end if**
$T_{A_t}$ += 1;
$\hat{\mu}_{A_t} = \frac{\hat{\mu}_{A_t} \times (T_{A_t} - 1) + r(t)}{T_{A_t}}$ ;
$reward_n$+= $\hat{\mu}_{A_t}$
$Regret_n = Regret_n + \max \hat{\mu}_i - \hat{\mu}_{A_t}$
**end for**
**Output:** The cumulative regret $Regret_n$
The cumulative reward $reward_n$

## 3. Empirical Investigations: Datasets and Experimental Design

### *3.1. Dataset Overview*

The data used in this experiment came from data sets MovieLens. These files contain 1,000,209 anonymous ratings of approximately 3,900 movies made by 6,040 MovieLens users who joined MovieLens in 2000 [10]. In this paper, the original time interval in the data set is cancelled, the scoring time is re-assigned according to the order of time, and each movie category is regarded as an arm, so there are a total of 18 arms.

Then all movie scores are classified according to the type of movie, and the scores of various movies at each time are obtained, which is the revenue function in the algorithm. And in the data set, the movie score ranges from 1 to 5, which means that the return of each round in the algorithm is between 1 and 5 points. After the data set is sorted out according to this method, the experiment can be started.

### *3.2. Methodological Blueprint for Experiments*

In the experiment of this paper, the arm k of the n-arm slot machine model is set to 18, that is, the number of movie types in the movie data set used. In addition, 5,000 epochs will be simulated in each experiment, and each epoch will pull the pull rod 500,000 times in the sequence of strategies, and the reward and regret values returned by each pull rod will be recorded. Because random numbers generated during the process lead to noisy and unstable experimental results, an average reward and regret of 5,000 epochs were eventually used for plotting. After the drawing is completed, the performance of the algorithm is compared according to the graph of accumulated reward and accumulated regret.

The comparison of the experimental results of the accumulated reward and accumulated regret of the epsilon- decreasing greedy algorithm and the fixed epsilon-greedy algorithm is shown in Figure 3 and Figure 4. The single experimental result of the epsilon- decreasing greedy algorithm is shown in Figure 5, in which the super-parameter value of the fixed epsilon-greedy algorithm is selected as 0.0001,0.001,0.01, 0.1,0.3,0.5,0.7,0.9, respectively. The super-parameter value of the epsilon-decreasing greedy algorithm is selected as 1. It can be seen from Figure 3, Figure 4 and Figure 5 that the epsilon- decreasing greedy algorithm can converge in a short time and obtain the maximum cumulative reward, while the fixed epsilon-greedy algorithm can only grow linearly without convergence. In conclusion, compared with the fixed epsilon-greedy algorithm, the epsilon- decreasing greedy algorithm can better solve the cold start problem of new users, and can converge in a short time and maintain a better recommendation effect.

DOI: 10.54254/2755-2721/54/20241140

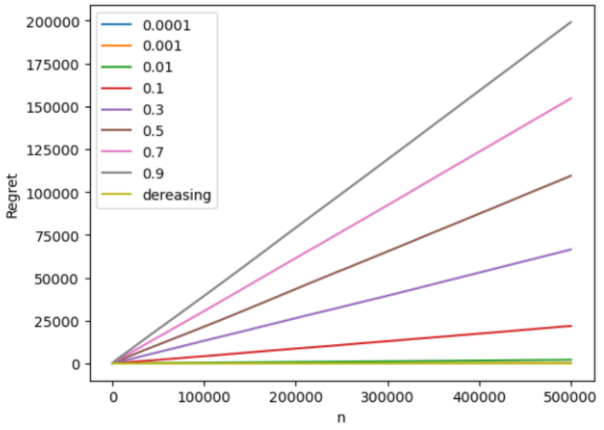


**Figure 3.** Cumulative regret comparison between the two algorithms (Photo/Picture credit: Original).

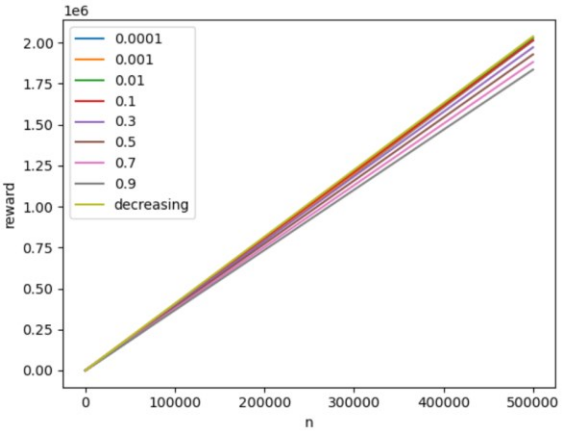


**Figure 4.** Comparison of cumulative reward between the two algorithms (Photo/Picture credit: Original).

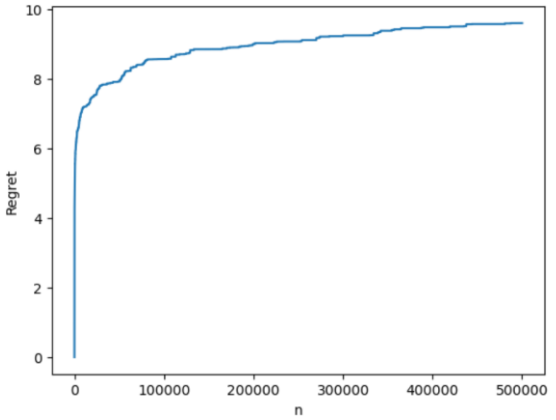


**Figure 5.** Cumulative regret diagram of the epsilon- decreasing greedy algorithm (Photo/Picture credit: Original).

Figure 6 and Figure 7 respectively compare the cumulative regret of the epsilon- decreasing greedy algorithm after changing the attenuation constant $c$ and the cumulative reward. It can be seen that although the smaller the $c$ is, the better the cumulative regret is, the smaller the c is, the worse the cumulative reward is. Therefore, it can be seen that properly increasing the size of $c$ and slowing down the decreasing rate of epsilon can obtain better revenue effects.

DOI: 10.54254/2755-2721/54/20241140

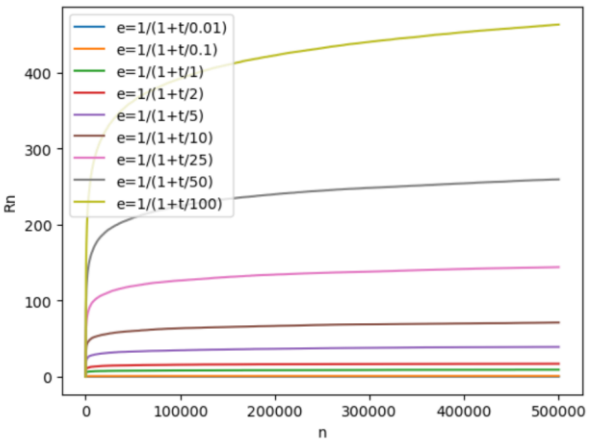


**Figure 6.** Cumulative regret comparison of different parameters c (Photo/Picture credit: Original).

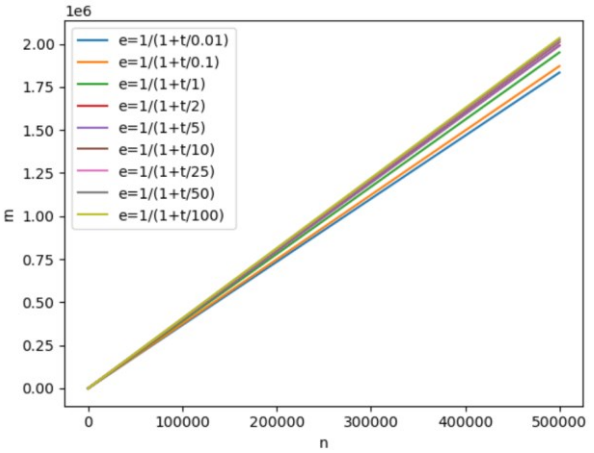


**Figure 7.** Cumulative reward comparison of different parameters c (Photo/Picture credit: Original).

Figure 8 and Figure 9 respectively show the cumulative regret comparison and cumulative reward comparison between the epsilon- decreasing greedy algorithm when the decreasing constant c is 1 and the fixed epsilon-greedy algorithm when epsilon is 0.01, ETC algorithm and UCB algorithm. It can be seen that the epsilon-decreasing greedy algorithm is better than the contrast algorithm in both indexes and converges quickly.

DOI: 10.54254/2755-2721/54/20241140

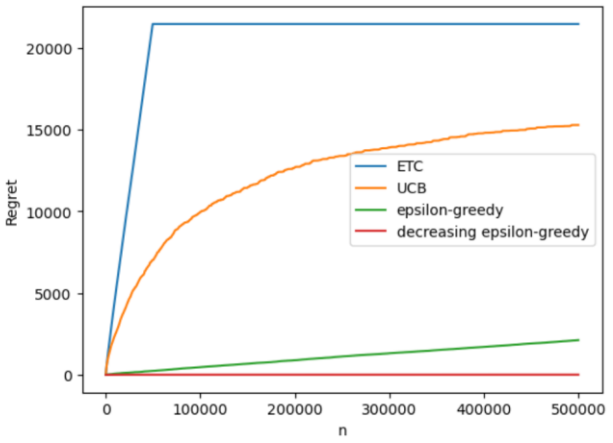


**Figure 8.** Cumulative regret comparison of the four algorithms (Photo/Picture credit: Original).

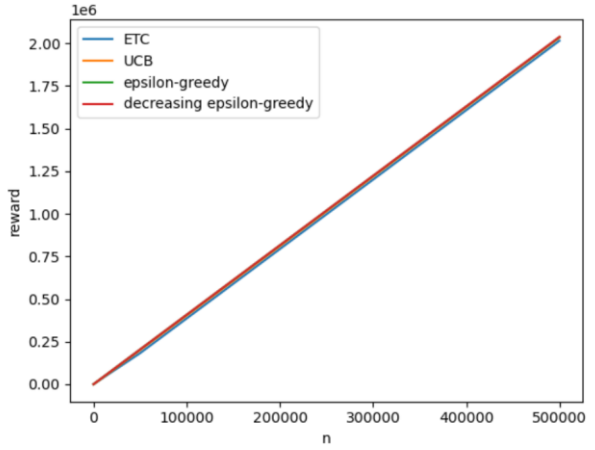


**Figure 9.** Cumulative reward comparison of the four algorithms (Photo/Picture credit: Original).

### *3.3. Results and Interpretation*

In the analysis of the above experimental results, it can be seen that the epsilon-decreasing greedy algorithm is convergent compared with the fixed epsilon-greedy algorithm before the improvement, and the comparison of all indexes is better than the original algorithm and the comparison algorithm. It can be seen that the improved algorithm proposed in this paper has better performance and effect in solving the problem of film recommendation cold start.

## 4. Encountered Challenges and Future Expectations

In this paper, while efforts were made to optimize the algorithm from various perspectives, constraints in time and data availability hindered the success of several optimization avenues. This has resulted in a lack of comprehensive content within the paper, and the final optimization outcomes didn't delve deeply enough to effectuate substantial improvements in efficiency. For more impactful and meaningful advancements, the following areas can be considered:

- Explore a broader spectrum of improvement angles, testing various epsilon decay methods and contrasting their efficacy.
- Implement a selection of efficiency evaluation techniques to facilitate better comparison.
- The current dataset employed is somewhat limited in scope. Future work should consider utilizing more expansive datasets to enhance the depth and applicability of findings.

DOI: 10.54254/2755-2721/54/20241140

## 5. Conclusion

Recommendation systems epitomize the technological advancements of the digital age. Serving as a robust countermeasure against the ubiquitous issue of information overload, these systems proficiently steer users towards pertinent information or products tailored to their preferences. Among the myriad challenges these systems face, the 'cold start' problem, especially in the realm of movie recommendations, remains a prominent hurdle. This phenomenon arises when new users are introduced to a platform, and the system lacks sufficient data to make precise recommendations. In this study, we've taken the foundational epsilon-greedy algorithm and have refined it, yielding an enhanced epsilon-decreasing greedy methodology. Our optimized approach is adept at offering recommendations to new users based on prevailing movie data as soon as they engage with a movie application. This strategy effectively mitigates the cold start conundrum inherent in film recommendation systems. Our empirical experiments manifest promising outcomes, validating the algorithm's efficacy. Moving forward, our research endeavors will explore a diverse array of epsilon decay techniques. The ultimate goal is to further amplify the algorithm's performance, ensuring that even in the face of the cold start challenge, users receive optimal movie recommendations.

## References

[1] Tan, K. (2019). Application and Implementation of Movie Recommendation Algorithm Based on Reinforcement Learning. Doctoral dissertation, Huazhong University of Science and Technology.

[2] Wang, S., Zhang, Y., Jiang, H., et al. (2018). Improved Epsilon-greedy algorithm for cold-start problem of new users. Computer Engineering, 44(11), 6.

[3] Guo, Y-H., & Deng, G-S. (2008). Hybrid Recommendation Algorithm of Item Cold-start in Collaborative Filtering System. Computer Engineering, 34(23), 3.

[4] Lattimore, T., & Szepesvári, C. (2020). Bandit Algorithms. Cambridge University Press.

[5] Harper, F. M., & Konstan, J. A. (2015). The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS), 5(4), Article 19.

[6] Alwahhab, A. B. A. (2020, November). Proposed Recommender System for Solving Cold Start Issue Using k-means Clustering and Reinforcement Learning Agent. In 2020 2nd Annual International Conference on Information and Sciences (AiCIS) (pp. 13-21). IEEE.

[7] Elena, G., Milos, K., & Eugene, I. (2021). Survey of multiarmed bandit algorithms applied to recommendation systems. International Journal of Open Information Technologies, 9(4), 12-27.

[8] Afsar, M. M., Crump, T., & Far, B. (2022). Reinforcement learning based recommender systems: A survey. ACM Computing Surveys, 55(7), 1-38.

[9] Huang, H., Huang, J., Feng, Y., Zhang, J., Liu, Z., Wang, Q., & Chen, L. (2019). On the improvement of reinforcement active learning with the involvement of cross entropy to address one-shot learning problem. PloS one, 14(6), e0217408.

[10] Huang, T., Li, M., & Zhu, W. (2022). ACP based reinforcement learning for long-term recommender system. International Journal of Machine Learning and Cybernetics, 13(11), 3285-3297.