ITM Web of Conferences **78**, 01027 (2025) https://doi.org/10.1051/itmconf/20257801027
CSEIT 2025

Enhancing Dynamic Movie Recommendations With User Expectation Ratings in Contextual Bandit Models

Weiye Sun

School of Information Science and Technology, ShanghaiTech University, Shanghai, 200126, China

Abstract. Modern movie recommendation systems face challenges such as dynamic personalization and real-time adaptability. Traditional methods like collaborative filtering and content-based recommendations struggle with dynamic user preferences and cold-start problems. Contextual multi-armed bandit (CMAB) algorithms offer a promising solution by balancing the exploration and exploitation trade-off while incorporating contextual information. This paper evaluates the performance of CMAB algorithms in dynamic movie recommendation scenarios using the MovieLens Beliefs Dataset 2024. The study employs Linear Upper Confidence Bound (LinUCB) and Contextual Thompson Sampling (CTS) algorithms, incorporating user historical data and expected ratings as context. Results show that CMAB algorithms significantly outperform traditional multi-armed bandits (MAB) in terms of cumulative regret and rating accuracy. Specifically, LinUCB achieves rating accuracies of 0.712, compared to 0.623 for the Upper Confidence Bound (UCB) algorithm, indicating a 10% improvement. The enhanced context with user expectation ratings further improves recommendation performance. This research demonstrates the effectiveness of CMAB algorithms in dynamic environments and provides insights for future recommendation system designs.

1 Introduction

Modern movie recommendation systems still face many challenges, including but not limited to dynamic personalization, scalability, and real-time adaptability. Traditional movie recommendation methods, such as collaborative filtering recommendation and content-based recommendation, have more limitations and fail to effectively handle dynamic user preferences, cold-start problems, and instant feedback [1, 2]. Currently, in recommendation scenarios, MAB algorithms are a promising tool for balanced exploration and exploitation, but classic bandit frameworks exhibit a large number of limitations, since they ignore contextual information, such as user preferences and historical browsing history, leading to a negative impact on recommendation results [3]. In addition, they cannot effectively respond to environments where changes in user preferences fluctuate significantly. These shortcomings highlight the need for more sophisticated frameworks, and it is particularly important to utilize contextual information while maintaining computational efficiency [4].

sunwy12022@shanghaitech.edu.cn

The main research goal of this paper is to evaluate the performance of CMAB algorithms in a real-world movie recommendation scenario. Compared to traditional bandits, CMABs are able to optimize decision making by adapting recommendation methods based on contextual factors such as users' movie-viewing hours and genres [5]. However, their actual functionality has not been fully investigated and proven in dynamic real-world environments. This paper aims to analyse CMABs using real-world datasets, discuss whether it can make up for the limitations of classic bandits, and focus on evaluation metrics such as minimization of cumulative regrets and users' expected ratings. In this paper, the author designs a framework based on the CMAB algorithm, which collects contextual information while maintaining computational efficiency. The author processes the dataset and sets the user's historical movie-browsing information and predicted user ratings as the context; the author uses the movie genres as the arm and the user's actual ratings as the reward. In particular, this paper introduces the gap between eliciting and predict-rating as a new evaluation metric, which provides a more intuitive representation of the recommendation effect.

2 Related Work

Past movie recommendation systems mainly use filtering techniques to filter and identify similar features based on content and continuously recommend movies that overlap with these features [2]. This approach has a certain amount of error, as the user may not be able to enjoy all the movies with the same features. Moreover, it fails to make full use of users' information. Much of the current research in the field of recommender systems is beginning to focus on multi-armed bandit algorithms. The classical bandit algorithms are UCB and Thompson Sampling (TS), which lay the foundation for the exploration-exploitation trade-off in recommendation systems. Although these methods are effective in fixed environments, they lack the use of contextual data, which limits their applicability to dynamic personalized recommendations. A research team implements a framework based on classic bandit algorithms, including epsilon-greedy and UCB, and applies it to evaluation on ad click-through rates [3]. However, this work is limited to a stationary environment. Hence, in realistic scenarios, the reward distribution may fluctuate rapidly, making it difficult to deploy this model. Another group of researchers analyzes Thompson Sampling's effect on recommendations for healthcare [6]. It is shown that despite Thompson Sampling's potential ability in generalization, the algorithm does not address some of the biases in real-world data, while missing verification in a variety of real-life dynamic healthcare environments.

To address this issue, contextual slot machine algorithms such as LinUCB and CTS have been introduced. LinUCB extends UCB by modelling rewards as linear functions of contextual features, making arm selection feature-driven. CTS integrates context into Bayesian sampling, thus providing stability against uncertainty. In a paper, the authors design a bandit-based personalized recipe recommendation in which the LinUCB algorithm is used to train the weight parameters [7]. This system is able to balance users' nutritional intake and flavour preferences. In a paper, the author develops strategies for personalized learning action selection using CTS and successfully designs personalized learning actions that improve students' scores [8]. In addition, some researchers use contextual bandit algorithms to improve ad click-through rates when studying personalized ad recommendations [9]. This method improves decision-making by focusing on users' historical data and yields recommendations that can be applied to various scenarios.

3 Methodology

Fig. 1 is the framework of the whole experiment. The process begins with data preprocessing, which involves four key components: base context, enhanced context, arm, and reward. Following data preprocessing, the experiment proceeds to algorithm implementation. In this section, LinUCB and CTS are applied. The results obtained are then compared with algorithms from a control group, which includes traditional MAB, contextual MAB with basic context, and contextual MAB with enhanced context. Finally, the outcomes are visualized using three evaluation metrics: cumulative regret, ratings' accuracy, and cold-start performance.

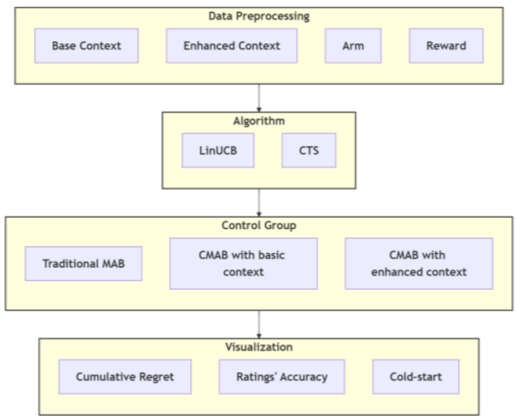


Fig. 1 Framework of the whole experiment (Photo credit: Original)

3.1 Data Preprocessing

The dataset is chosen from Grouplens' MovieLens Beliefs Dataset 2024, which investigates user ratings of watched movies and expected ratings of unwatched movies [10]. In this paper, the dataset was pre-processed to focus on extracting three key factors: movie genre, user ratings, and expected user scores. They are separately used as arms, rewards, and context.

3.2 Algorithms

This paper formulates the movie recommendation task as a contextual multi-armed bandit algorithm, where movie genres serve as the arms. User behaviour, movie genres, and user expectation ratings are encoded as the context. The author sets the reward as the actual user rating with a confidence weight. At each timestep, the algorithm observes the context, selects the optimal arm, and gains a reward.

The main concept of LinUCB is to model user-movie interaction rewards based on linear regression, quantify uncertainty through confidence intervals to achieve the exploration-exploitation trade-off, and dynamically select movies by calculating confidence upper bounds [7].

Specifically, LinUCB represents both users and movies as feature vectors, capturing various contextual attributes such as user historical ratings and movie genres. For each user-movie pair, a context vector is constructed by concatenating the user and movie feature vectors, denoted as $x_{u,i} = [x_u; x_i]$. The algorithm initializes a parameter matrix A as a scaled identity matrix $A = \lambda I$ and a vector b as a zero vector, where λ is a regularization parameter. At each time step, the estimated reward $\widehat{r_{u,i}}$ for each user-movie pair is calculated using the current parameter estimate $\theta = A^{-1}b$ through the linear relationship $\widehat{r_{u,i}} = x_{u,i}^{\top}\theta$. To quantify the uncertainty and balance exploration and exploitation, the upper confidence bound is computed as formula 1, where α is a hyperparameter controlling the trade-off.

$$UCB_{u,i} = \widehat{r_{u,i}} + \alpha\sqrt{x_{u,i}^{\top}A^{-1}x_{u,i}} \tag{1}$$

The algorithm then selects the movie i^* that maximizes this UCB. After observing the actual reward r_{u,i^*}, the parameter matrix and vector are updated as formulas 2 and formula 3 respectively, refining the model iteratively to better capture the user-movie interactions and adapt to changing preferences dynamically.

$$A \leftarrow A + x_{u,i^*}x_{u,i^*}^{\top} \tag{2}$$

$$b \leftarrow b + r_{u,i^*}x_{u,i^*} \tag{3}$$

The advantages of the LinUCB algorithm are higher computational efficiency and strong interpretability of features with linear relationships, hence, LinUCB is suitable for dealing with user feedback with higher certainty and recommendations for users with stable interests.

The core idea of CTS is based on a Bayesian probabilistic framework, which realizes probabilistic exploration through parameter posterior distribution sampling and balances exploration and exploitation. The algorithm can better deal with nonlinear relationships and is suitable for capturing dynamic changes in user interest. It is also applicable to new users' cold-start scenarios [11].

3.3 Evaluation Metrics

In order to fully evaluate the performance of contextual MABs in dynamic movie recommendation scenarios, the author employs several key evaluation metrics. These metrics are designed to assess the effectiveness, accuracy, and adaptability of the algorithm in real-world applications.

3.3.1 Cumulative Regret Curve

The cumulative regret curve is a classic metric for evaluating the performance of MAB algorithms. It measures the difference between the expected optimal reward and the actual reward over time, reflecting how well the algorithm balances exploration and exploitation. The cumulative regret $R(t)$ at time t is calculated as formula 4, where $r_{\mathrm{opt},\tau}$ is the expected optimal reward at time step τ, and $r_{\mathrm{actual},\tau}$ is the actual rating given by the user for the i-th recommended movie.

$$R(t) = \sum_{\tau=1}^{t}\left(r_{opt,\tau} - r_{actual,\tau}\right) \tag{4}$$

A lower cumulative regret value indicates that the algorithm has made better decisions over time, effectively reducing the opportunity cost of not selecting the optimal arm at each time step. Plotting cumulative regret curves for multiple time steps yields a visual assessment of the overall performance of the MAB algorithm.

3.3.2 Rating Accuracy

Another important metric used in this paper is rating accuracy, which measures how well a user's actual rating agrees with the user's expected rating. This metric is important because it is particularly relevant to dynamic movie recommendations and can directly reflect the algorithm's ability to understand and predict user preferences. The rating accuracy is calculated as formula 5, where N is the total number of recommendations, $r_{\text{expected},i}$ is the expected rating for the i-th recommended movie, and $r_{\text{actual},i}$ is the actual rating given by the user for the i-th recommended movie.

$$Accuracy = 1 - \frac{1}{N}\sum_{i=1}^{N}\left|r_{expected,i} - r_{actual,i}\right| \tag{5}$$

The smaller the difference between the expected and actual ratings, the higher the accuracy, indicating that the algorithm effectively captured the user's expectations and preferences. This metric is particularly useful for evaluating the performance of the algorithm in situations where user preferences may change over time.

3.3.3 Cold-start Problem

The cold-start problem is a common challenge in recommendation systems. It especially focuses on scenarios where algorithms need to deal with new users or new movies with limited historical data. When evaluating the cold-start performance of contextual MAB algorithms, this paper focuses on the speed at which the algorithm adapts to a new user or a new movie and provides it with accurate recommendations. The author measures the initial performance of the algorithm by observing the algorithm's rating accuracy and cumulative regret with the fewest historical interactions. Better algorithms are able to learn more quickly from limited data and provide more meaningful recommendations, thus minimizing the negative impact of cold-start problems on the overall user experience.

4 Results and Analysis

4.1 Setup

In the setup of MAB problems, there are basically n > 1 arms, and each pull of an arm generates a random reward with an unknown distribution, while the goal is to find the arm with the highest expected reward, which is defined as the optimal arm. The algorithm will trade off between choosing the random arm and choosing the known optimal arm. This exploration-exploitation trade-off allows the algorithm to converge over time to choosing the optimal arm [7].

In the scenario of a movie recommendation system, this paper sets movie genre as the arm and user ratings as the reward. The author designs four experiments with the aim of comparing the performance between MAB, CMAB, and context-enhanced CMAB under various conditions.

4.2 Dataset

The dataset chosen for this paper is the MovieLens Beliefs Dataset 2024 from Grouplens. Grouplens provides research data in a variety of areas such as recommendation systems, online communities, and digital libraries. This paper aims to study the application of CMAB in the field of movie recommendation systems, so this database is chosen. It provides users' movie recommendations, ratings, and expected ratings based on the movie recommendation service MovieLens. In the data preprocessing session, this paper cleanses the data from this dataset, categorizes the movie genres, merges the user rating data, and creates user history features.

4.3 Experiment

4.3.1 Classic MAB vs. Contextual MAB

In this section, the author compares the classical MAB algorithm with the CMAB algorithm in a dynamic movie recommendation scenario. The classic MAB selects the arm only by the user's historical ratings, while the CMAB has additional contextual information to help select the arm. Fig. 2 shows the cumulative regret curves of the UCB algorithm, the TS algorithm, and the LinUCB algorithm. It can be seen that LinUCB better minimizes the cumulative regret and significantly outperforms the two classical MAB algorithms.

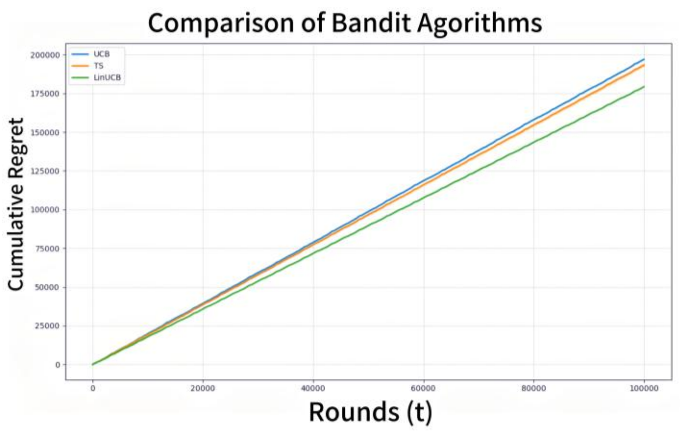


Fig. 2 Comparison of bandit algorithms (Photo credit: Original)

Table 1 shows the rating accuracy of UCB, TS, LinUCB, and the CTS algorithm. It can be seen that the accuracy of the two CMAB algorithms is about 10% higher than that of the MAB algorithms, which suggests that the CMAB algorithms are better at predicting user preferences. The difference in accuracy between the two CMAB algorithms is small, and the CTS algorithm is slightly better than the LinUCB algorithm.

Table 1 Rating accuracy of different bandit algorithms

Algorithm	Accuracy	Std Dev
UCB	0.623	0.012
TS	0.656	0.011
LinUCB	0.712	0.009
CTS	0.734	0.008

4.3.2 Enhanced Contextual MAB

In this section, the aim of the experiment is to investigate the effect of enhanced context on the performance of the CMAB algorithm. The author adds the user's expected ratings as an additional context feature and compares them with the algorithm in the base context scenario. Fig. 3 shows the cumulative regrets of LinUCB in the base context and the enhanced context. It can be concluded that the cumulative regret of the algorithm is reduced after the introduction of the user expectation ratings. Compared to the base context, LinUCB in the enhanced context shows an improvement in performance.

Fig. 3 Comparison of LinUCB in base context and enhanced context (Photo credit: Original)

Table 2 shows the rating accuracies of base LinUCB, base CTS, enhanced LinUCB, and enhanced CTS. It can be seen that the CMAB algorithm with enhanced context will predict user preferences better than the base context.

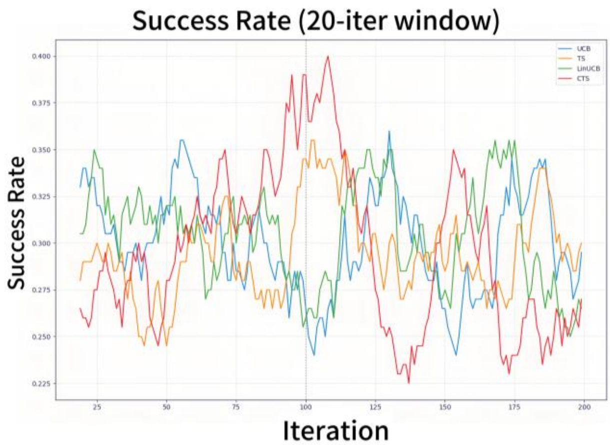
Table 2 Rating accuracy of different CMABs

Algorithm	Accuracy	Std Dev
LinUCB (Base)	0.7123	0.0214
LinUCB (Enhanced)	0.8231	0.0187
CTS (Base)	0.7765	0.0252
CTS (Enhanced)	0.8812	0.0169

4.3.3 Cold-start Scenario

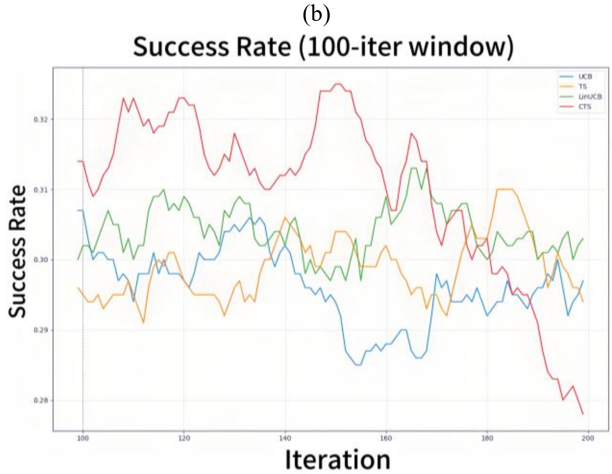
In this section, the author filters new users with the least amount of historical information to simulate a cold-start scenario. Fig. 4 shows the comparison of success rates under different window sizes for observing the exploration efficiency, stability, and convergence speed of the algorithms. It can be found that the CTS algorithm rises the fastest and has better cold-start ability in the early stage with window sizes of 20 and 50, while the UCB and LinUCB algorithms are relatively stable. In the later stages, with a window size of 100, the LinUCB and CTS algorithms have significantly higher success rates than the MAB algorithms. The CTS fluctuates slightly, the LinUCB has the best stability, and the UCB has the worst performance.

 https://doi.org/10.1051/itmconf/20257801027



(a)

(b)

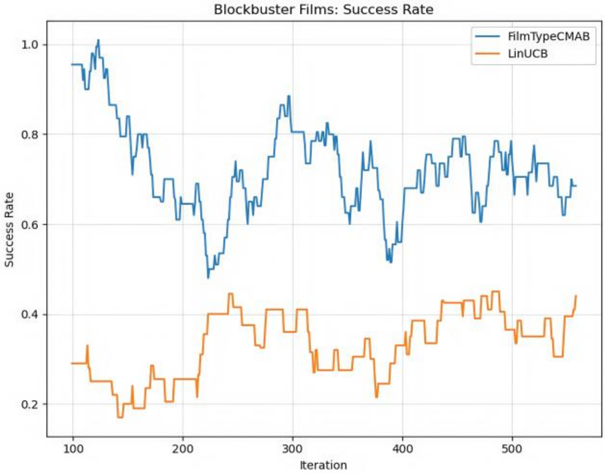


(c)

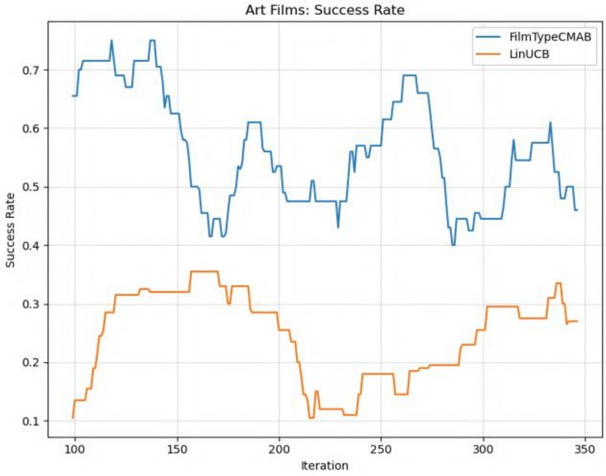
Fig. 4 (a) Window of size 20 (Photo credit: Original), (b) Window of size 50 (Photo credit: Original), (c) Window of size 100 (Photo credit: Original)

4.3.4 Recommendation Effectiveness for Different Movie Genres

The experiment in this section aims to evaluate the difference in the efficiency of the CMAB algorithms in recommending blockbuster and art movies. The author designs the movie genre-aware CMAB algorithm based on the LinUCB algorithm, which maintains separate LinUCB models for commercial and literary movies and uses a larger exploration factor for literary movies. Fig. 5 shows the comparative success rate curves of commercial and literary films.



(a)



(b)

Fig. 5 (a) Success rate for blockbuster films (Photo credit: Original), (b) Success rate for art movies (Photo credit: Original)

Table 3 reflects the success rate and diversity of different algorithms for different movie genres. By comparison, it can be found that the success rate of movie-aware CMAB is substantially improved. In terms of recommendation diversity, literary films are generally higher than commercial films.

Table 3 Success rate and diversity of different CMABs

Algorithm	Film Type	Success Rate	Diversity
movie genre-aware CMAB	Blockbuster	0.7033	0.3017
movie genre-aware CMAB	Art	0.5394	0.9256
LinUCB	Blockbuster	0.3819	0.2478
LinUCB	Art	0.2375	0.8561

5 Conclusion

Through the four experiments and analysis, the author finds that CMAB algorithms, specifically LinUCB and Contextual Thompson Sampling, significantly outperform traditional multi-armed bandit algorithms in dynamic movie recommendation scenarios. The incorporation of contextual information, such as user historical data and expected ratings, effectively reduces cumulative regret and improves rating accuracy by about 10% compared to traditional methods. The enhanced context with user expectation ratings further improves performance, with CTS showing slightly better accuracy than LinUCB. In cold-start scenarios, CTS demonstrates superior adaptability for new users or movies, while LinUCB exhibits better stability in the long run. Additionally, the genre-aware CMAB algorithm tailored for different movie types achieves higher success rates and recommendation diversity, especially for art films.

This paper fills the gap in evaluating CMAB algorithms in dynamic movie recommendation systems and highlights their potential for real-world applications. It provides valuable insights for other researchers and practitioners on how to utilize contextual information to improve recommendation performance. The findings suggest that incorporating user expectation ratings as an additional context feature can effectively enhance recommendation accuracy and help predict user preferences better. The study also demonstrates the importance of considering diverse movie genres in recommendation systems to better cater to different users.

Despite the promising results, this study has some limitations. The CMAB algorithms used in the paper are limited to LinUCB and CTS. In the future, research on more complex contextual algorithms to cope with dynamic user preferences and non-linear relationships can be considered. In terms of evaluation metrics, user click-through rate and retention rate can be further introduced to more comprehensively assess how well the algorithms are performing. Moreover, in the section of enhanced context, considering whether there are other contexts that can strengthen the performance of the algorithms remains an important direction for future work.

References

1. Karthikeyan P., Tejasvini C.: 'Review of movie recommendation system'. Proc. 8th Int. Conf. Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2022, pp. 1538-1543
2. P. D S., S. A, P. M, C. S, L. S.K.: 'Personalized movie recommendations: A comprehensive review and analysis'. Proc. 10th Int. Conf. Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2024, pp. 2042-2045

3. Mambou E.N., Woungang I.: 'Bandit algorithms applied in online advertisement to evaluate click-through rates'. Proc. IEEE AFRICON, Nairobi, Kenya, 2023, pp. 1-5
4. Taj N., Varun M.H., Navya V.: 'Advanced content-based movie recommendation system'. Proc. 5th Int. Conf. Smart Electronics and Communication (ICOSEC), Trichy, India, 2024, pp. 1693-1698
5. Li H., Wang M., Zhang J., Shi T., Khamis A.: 'A contextual multi-armed bandit approach to personalized trip itinerary planning'. Proc. IEEE Int. Conf. Smart Mobility (SM), Niagara Falls, ON, Canada, 2024, pp. 55-60
6. Dixit K.K., Verma D., Muthuvel S.K., Laxminarayanamma K., Kumar M., Srivastava A.: 'Thompson sampling algorithm for personalized treatment recommendations in healthcare'. Proc. Int. Conf. Artificial Intelligence for Innovations in Healthcare Industries (ICAIIHI), Raipur, India, 2023, pp. 1-6
7. Subhedar A., Bhutada P., Khanna A.A., Gupta R.K., Diddigi R.B.: 'Personalized diet recommendation system using bandits'. Proc. IEEE Int. Conf. Future Machine Learning and Data Science (FMLDS), Sydney, Australia, 2024, pp. 481-486
8. Manickam I., Lan A.S., Baraniuk R.G.: 'Contextual multi-armed bandit algorithms for personalized learning action selection'. Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 2017, pp. 6344-6348
9. S. G. S., et al.: 'Dynamic personalized ads recommendation system using contextual bandits'. Proc. Int. Conf. Intelligent Systems for Communication, IoT and Security (ICISCoIS), Coimbatore, India, 2023, pp. 339-344
10. 'MovieLens Beliefs Dataset 2024 | GroupLens', https://grouplens.org/datasets/movielens/ml_belief_2024/, accessed 25 April 2025
11. Gutowski N., Amghar T., Camp O., Chhel F.: 'Context enhancement for linear contextual multi-armed bandits'. Proc. IEEE 30th Int. Conf. Tools with Artificial Intelligence (ICTAI), Volos, Greece, 2018, pp. 1048-1055