Personalized Recommendation System Based on Deep Reinforcement Learning

Yuning Wang

International Business School, Beijing Foreign Studies University, Beijing, China

Abstract. With the rapid development of Internet technology, personalized recommendation systems have become increasingly important in fields such as e-commerce, social media, and online entertainment. However, accurately capturing the dynamically changing interests of users and providing high-quality recommendations still remain the major challenges in the field of recommendation systems. This paper proposes a personalized recommendation system based on deep reinforcement learning, and adopts the Actor-Critic architecture to optimize the recommendation strategy. This paper conducts a detailed analysis of the impacts of different reward normalization strategies, experience replay, and state representation strategies on the performance of the model. Through experiments conducted on the MovieLens ml-1m dataset, we have verified the effectiveness of this method. This method not only outperforms existing methods in terms of total reward, Q-loss, and precision, but also demonstrates better stability and scalability when dealing with large-scale data and complex user behaviour patterns. It also explores how to further improve the accuracy of the recommendation system and user satisfaction, providing new perspectives and methods for future research on recommendation systems.

1 Introduction

In recent years, personalized recommendation systems have been increasingly significant. The essential goal is to recommend the most relevant and appealing content to users based on their historical behaviour and preferences. However, accurately capturing users' dynamic interests and providing high-quality recommendations remain the major challenges in this field. For example, users' preferences for item sequences can change dynamically with real-time situations. This requires recommendation systems to have the ability to adapt to the context immediately, highlighting the necessity of modelling dynamic user interests [1].

The history of recommendation systems dates back to early collaborative filtering (CF) and content-based recommendation. However, these traditional methods are limited in handling dynamic interests and the cold start problem, etc. For instance, Abdolmaleki and Rezvani argued that the lack of sufficient user interest information in the initial stage is a key issue for collaborative filtering [2]. Wei et al. proposed a multi-stage dynamic Bayesian network (MS-DBN), which relies on static probability models to capture user shopping

202220102023@bfsu.edu.cn

stages and thus struggles to capture complex temporal features [3]. Bi et al. developed a consumer acquisition model that, despite considering network externalities, is based on static network assumptions and cannot adapt to dynamic changes in user behaviour [4]. Furthermore, Adamopoulos proposed a graph-structured debiasing method, which is not suitable for dynamic user-item interaction scenarios [5]. These studies indicate that traditional methods face insufficient modelling capabilities in dynamic environments and urgently need a more flexible technical framework.

In recent years, deep learning technologies have provided new directions. Models such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Graph Neural Networks (GNN) are capable of processing complex inputs and temporal data, significantly enhancing the system's ability to understand context. For example, Farashah et al. improved link prediction recommendation efficiency by combining deep neural networks with an improved Friendlink algorithm [6]. Farahani et al. proposed a dynamic personalized recommendation system that tracks interest drift in real time through adaptive user profile learning [7].

Deep Reinforcement Learning (DRL) has also been applied to recommendation systems. By simulating the interaction between an agent and its environment to learn optimal policies, DRL provides a new paradigm for dynamic recommendations. For example, Chen et al. proposed a Transformer-DRL framework that captures complex temporal patterns to dynamically optimize recommendations [8]. Fu et al. demonstrated that their multi-policy DRL system, which dynamically adjusts strategies for cold-start users, significantly improves recommendation coverage [9]. Kokkodis and Ipeirotis developed a DRL-based career path recommendation framework that uses Markov decision processes to dynamically plan skill acquisition paths, successfully increasing freelancers' income by 22% and skill diversity by 47% [10]. Bukhari et al. showed that their context-aware Actor-Critic model, can significantly improve the precision of DRL recommendation systems [11]. However, Pang et al. pointed out that existing DRL methods often suffer from unstable model convergence due to sample imbalance and action space explosion [12]. These studies indicate that although DRL can effectively address the dynamic changes in user interests, its performance is highly dependent on the design of the reward function and the strategy for environment modelling.

Research in the field of movie recommendation continues to evolve. Early work focused on improving recommendation efficiency and accuracy, while recent studies have paid more attention to capturing dynamic interests and optimizing diversity. For example, Ambikesh et al. proposed the GOA-Kmeans model, which improves recommendation accuracy by calibrating user preferences in real time [13]. Yadav et al. and Sahu et al. respectively used content analysis and deep learning to predict movie audience groups, verifying the effectiveness of multi-feature fusion [14][15]. In addition, Bi et al. showed that dynamically adjusting incentive strategies can simultaneously increase platform utility and user retention [4]. Huang et al. proposed a semantic similarity model based on multiple knowledge resources and applied it to a movie recommendation system [16]. By combining the knowledge resources of Wikipedia and WordNet, the relevance and accuracy of recommendations are improved. These findings provide insights for the fairness, diversity, and dynamic adaptability of movie recommendation systems, but further exploration is needed on achieving multi-objective optimization through the DRL framework.

This paper proposes a personalized recommendation system based on Deep Reinforcement Learning (DRL), which optimizes the recommendation strategy using the Actor-Critic architecture. This method achieves better performance than existing methods on the MovieLens ml-1m dataset. The focus of this project is to analyse the impacts of different reward normalization strategies, as well as experience replay and state representation strategies on the model performance, and to explore how to further improve

the accuracy of the recommendation system and user satisfaction. Most of the existing studies focus on specific reward functions or strategies. This project provides more comprehensive guidance for the design of recommendation systems by systematically analysing multiple strategies.

2. Methodology

2.1 Dataset

This paper utilizes the MovieLens ml-1m dataset, as detailed in Table 1, a widely used public dataset for recommendation systems. This dataset contains 1 million rating records of 3,900 movies by 943 users. It is divided into three files: the user information file (users.dat), the movie information file (movies.dat), and the rating information file (ratings.dat). The user information file includes basic information of users, such as gender, age, and occupation. The movie information file contains basic details of movies, like titles and genres. The rating information file consists of users' rating records for movies.

Table 1 MovieLens ml-1m Dataset

Column Name	Description	Data Volume
UserID, Gender, Age, Occupation, Zip-code	Basic information of users	943 users
MovieID, Title, Genres	Basic information of movies	3,900 movies
UserID, MovieID, Rating, Timestamp	Users' rating records for movies	1 million records

The preprocessing phase of this study entails importing datasets into Pandas DataFrames, calculating the count of ratings for each user, creating a dictionary that records movies rated by each user along with their scores, mapping movie IDs to their titles for ease of analysis, and partitioning user data into training and test sets at a ratio of 80:20 to facilitate subsequent model development and assessment.

2.2 Neural Network Design

The design includes two main components: the Actor and Critic networks, each carefully designed to handle different aspects of the decision-making process.

Figure 1 depicts the Actor network which is structured to determine the optimal action given a state. It consists of an input layer that receives the state representation, followed by two hidden layers with 128 neurons each, designed to capture complex patterns within the data. The output layer, containing 100 neurons, produces the action vector, which is passed through a tanh activation function to ensure the output remains within a bounded range.

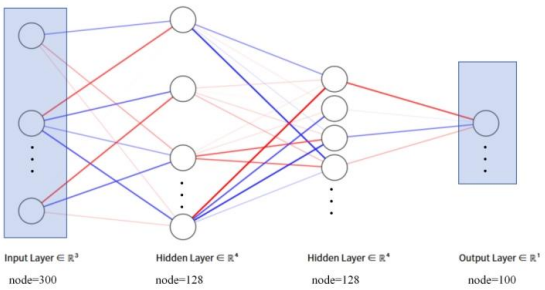

Fig. 1 Actor Network

Figure 2 illustrates the Critic network evaluates the quality of the state-action pairs. It features an input layer that concurrently receives the state and action vectors. This is followed by two hidden layers, each comprising 128 neurons, to process the combined input and assess the potential value. The final output layer, containing a single neuron, outputs the estimated Q-value, indicating the expected utility of taking a particular action in a given state.

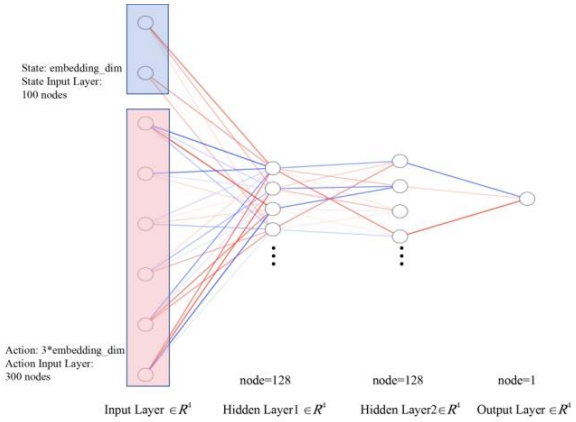

Fig. 2 Critic Network

2.3 Algorithmic Enhancements

Prioritized Experience Replay (PER): This technique adjusts the sampling probability of experiences based on their temporal difference (TD) error. By prioritizing experiences with higher TD errors, PER accelerates learning from critical transitions, thereby improving the efficiency of the algorithm.

Concatenated Embeddings: To enrich the feature representation, embeddings from different sources are concatenated. This approach integrates diverse information into a unified vector, enhancing the model's ability to discern intricate relationships within the data.

Reward Normalization: The scale of rewards can significantly influence the learning process. Normalizing the rewards mitigates the impact of varying reward magnitudes, leading to a more stable and efficient training regime.

Tanh-based Reward Normalization: Further enhancing reward processing, the tanh function is applied to normalize rewards, compressing them into a range of [-1, 1]. This

method effectively prevents extreme reward values from dominating the learning process, promoting a balanced update mechanism.

Figure 3 illustrates the workflow of a Deep Deterministic Policy Gradient (DDPG) algorithm with enhancements, showcasing the interaction between the Actor and Critic networks, experience replay memory, and the training process involving reward normalization and prioritized sampling.

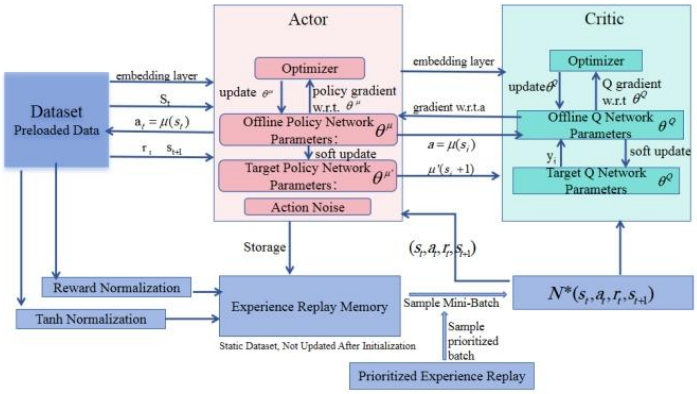

Fig. 3 Technical Flow Chart

2.4 Training Parameters

The model undergoes training over 500 and 1000 episodes, as specified in Table 2, with each episode involving interactions with the environment, learning updates to the neural networks, and refinement of the policy and value functions. The training process leverages a batch size of 32, ensuring that the gradient estimates are robust and representative of the overall dataset distribution.

Table 2 Model Parameter Description

Parameter Name	Parameter Value	Description
embedding_ dim	100	Dimension of the user and movie embedding vectors
hidden_dim	128	Dimension of the hidden layer
tau	0.001	Discount factor for soft target network updates
state_size	10	Number of movies included in the state representation
batch_size	32	Number of samples used per update
discount_fact or	0.9	Discount factor for rewards
epsilon	1.0	Initial exploration rate
epsilon_deca y	0.0001	Exploration rate decay rate
std	1.5	Standard deviation of action noise

3 Results

3.1 Figures

Figures 4 and 5 illustrate the progression of total rewards for various strategies over 500 and 1000 training epochs, respectively. Both figures demonstrate a general trend of increasing total rewards, which stabilize as training progresses. Notably, the strategy denoted as "norm_reward" exhibits superior performance in terms of total rewards in both the 500 and 1000 epoch analyses, indicating its robustness and effectiveness over extended training periods.

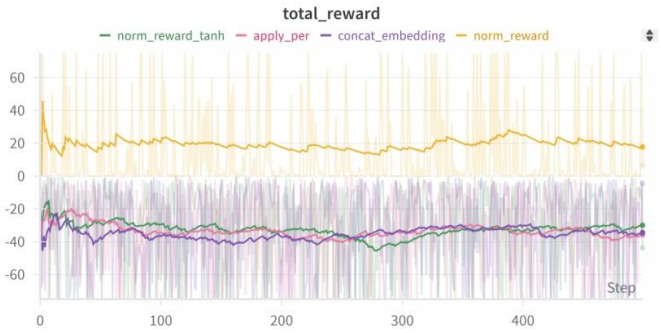

Fig. 4 Total Reward Comparison Across Strategies (500 Epochs)

Fig. 5 Total Reward Comparison Across Strategies (1000 Epochs)

Figures 6 and 7 depict the Q loss trajectories for different strategies across 500 and 1000 training epochs. A consistent pattern observed in both figures is the initial high Q loss that rapidly decreases and stabilizes as training advances, signifying improved prediction accuracy of the models. The strategy "norm_reward-tanh1000" in the 1000 epoch analysis shows the lowest Q loss, highlighting its enhanced predictive precision.

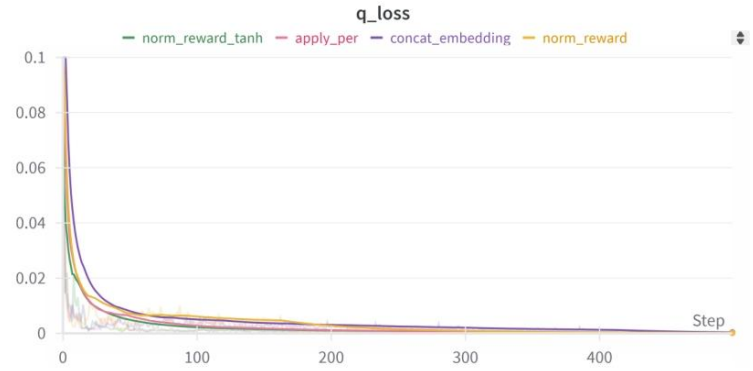

Fig. 6 Q-Loss Comparison Across Strategies (500 Epochs)

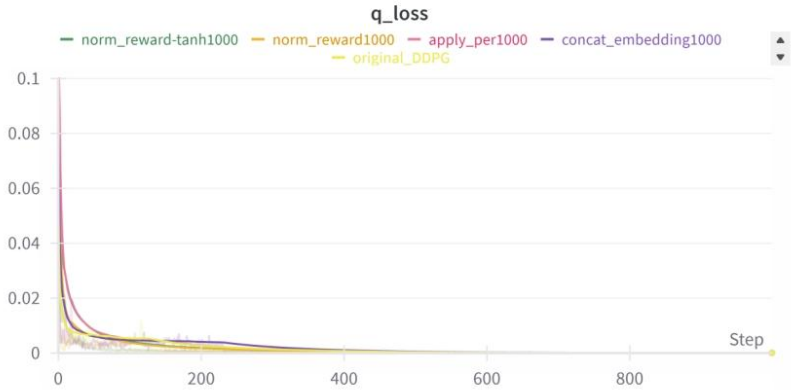

Fig. 7 Q-Loss Comparison Across Strategies(1000 Epochs)

Figures 8 and 9 track the mean action values for the strategies throughout 500 and 1000 training epochs. The data in both figures suggest that initial fluctuations in mean action values diminish over time, leading to a more consistent recommendation behavior. The "norm_reward-tanh1000" strategy, as shown in the 1000 epoch analysis, displays a significant stabilization in mean action values, underscoring its consistency in long-term training.

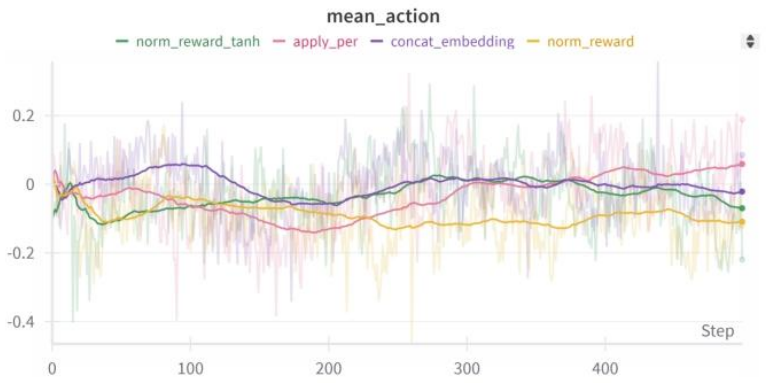

Fig. 8 Mean Action Comparison Across Strategies (500 Epochs)

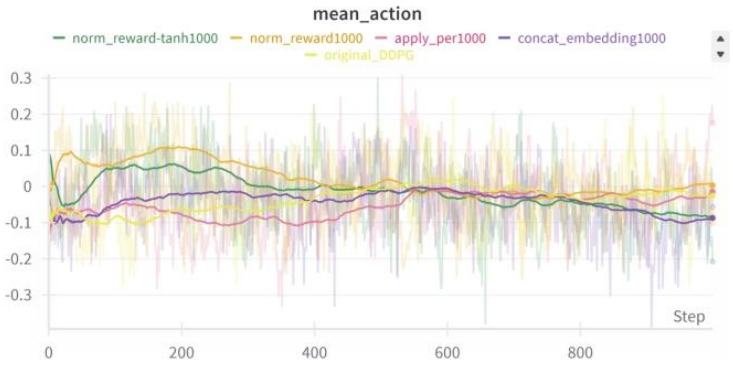

Fig. 9 Mean Action Comparison Across Strategies (1000 Epochs)

Figures 10 and 11 illustrate the epsilon values, which represent the exploration rates, for the strategies over 500 and 1000 training epochs. Both figures consistently show a decline in epsilon values as training progresses, aligning with the expected behavior of the ε-greedy strategy where exploration diminishes and exploitation increases with a better understanding of the environment. The extended training in the 1000 epoch analysis further reduces epsilon, reflecting a more refined exploration-exploitation balance.

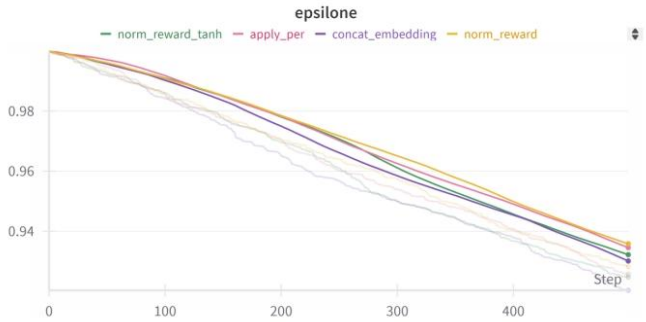

Fig. 10 Epsilon Comparison Across Strategies (500 Epochs)

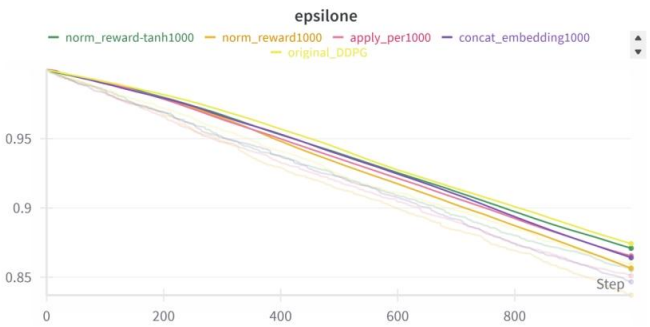

Fig. 11 Epsilon Comparison Across Strategies (1000 Epochs).

Figures 12 and 13 present the precision metrics for the strategies over 500 and 1000 training epochs. Both figures show fluctuations in precision during the initial training stages, which gradually stabilize. The strategy "norm_reward-tanh1000" in the 1000 epoch analysis stands out with improved precision, indicating its stability and reliability in long-term training scenarios.

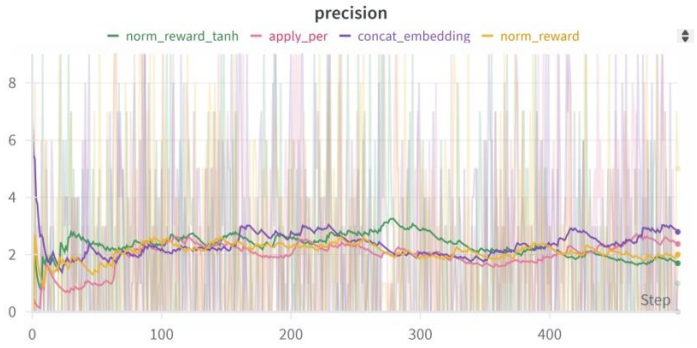

Fig.12 Precision Comparison Across Strategies (500 Epochs)

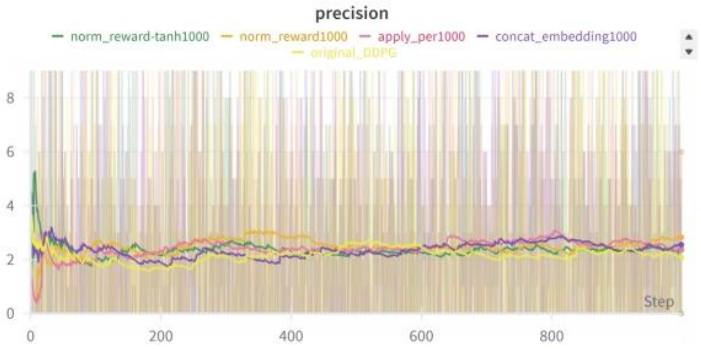

Fig. 13 Precision Comparison Across Strategies (1000 Epochs)

3.2 Results Analysis

Table 3 summarizes the findings from Figures 2-6, comparing four strategies in a DRL-based recommendation system. It evaluates their impact on total reward, Q-loss, precision, mean action, and exploration rate.

Table 3 Summary of Strategy Performance

Metric/Strategy	norm_reward	norm_reward_tanh	apply_per	concat_embedding
Total Reward	High stability, highest long-term	High fluctuation, later adaptation	Mostly negative, unstable	Mostly negative, high fluctuation
Q-loss	Stable convergence, lowest	Fast convergence, slightly inferior later	High fluctuation, requires optimizatio	Highest, needs learning rate adjustment

			n	
Precision (Precision)	Best stability	Peak high, high fluctuation	High fluctuation, some high points	Early good, later unstable
Mean Action (Mean Action)	Stable later	Stable later	Least fluctuation, best stability	Not converged, needs optimization
Epsilon (Exploration Rate)	Gradually decreases	Gradually decreases	Gradually decreases	Gradually decreases

3.2.1 Impact of Different Reward Normalization Strategies

The experimental results indicate that different reward normalization strategies significantly affect model performance. The "norm_reward_tanh" and "apply_per" strategies perform better in terms of precision, while the "norm_reward" strategy shows superior performance in terms of total reward. This suggests that the choice of a reward normalization strategy should be based on specific application scenarios and optimization objectives.

3.2.2 Model Convergence and Stability

The experimental outcomes demonstrate that the model gradually converges during the training process, with both Q-loss and exploration rate stabilizing over time. This indicates that the model's predictive accuracy and stability improve as it learns. However, the model exhibits greater fluctuations during the initial training phase, which may necessitate further adjustments to model parameters and exploration strategies to enhance the model's convergence rate and stability.

Table 4 provides an analysis of suitable scenarios for each strategy in the context of a deep reinforcement learning-based recommendation system. It offers tailored recommendations for optimizing strategy performance based on specific application needs. Norm_reward and norm_reward_tanh excel in long-term and rapid exploration scenarios, respectively, while apply_per and concat_embedding are suited for quick learning and complex feature integration.

Table 4 Analysis of Suitable Scenarios for Each Strategy

Strategy	Suitable Scenarios	Improvement Suggestions
norm_reward	Long-term training scenarios requiring high total reward and stability (e.g., long-tail recommendations)	Combine with other strategies (e.g., norm_reward_tanh) to accelerate initial exploration
norm_reward_tanh	Scenarios needing rapid initial exploration or short-term high precision output (e.g., popular item recommendations)	Adjust the scaling parameters of the tanh function, optimize initial exploration strategy

apply_per	Scenarios requiring quick learning of key samples with less emphasis on stability	Introduce dynamically adjusted IS weights or a mixed uniform sampling buffer to mitigate phase-specific fluctuations
concat_embedding	Complex feature scenarios requiring rapid feature integration, but needing optimization for long-term stability	Combine regularization (e.g., LayerNorm), adjust embedding dimensions, or introduce attention mechanisms instead of simple concatenation

4 Conclusion

This paper demonstrates the potential application of deep reinforcement learning in personalized recommendation systems and proposes a recommendation model based on the Actor-Critic architecture. The experimental results show that this model achieves superior performance on the MovieLens ml-1m dataset compared to existing methods. Future work will focus on further optimizing the model structure, exploring larger datasets, and examining the model's zero-shot learning capabilities across various fields. Additionally, incorporating more user feedback could enhance the model's adaptability and accuracy.

References

[1] Liebman, E., Saar-Tsechansky, M., Stone, P., et al.: 'The right music at the right time: Adaptive personalized playlists based on sequence modeling,' MIS Quarterly, 2019, 43, (3), pp. 765–789
[2] Abdolmaleki, A., & Rezvani, M. H.: 'optimal context - aware content - based movie recommender system using genetic algorithm: a case study on MovieLens dataset', Journal of Experimental & Theoretical Artificial Intelligence,2024, 36, (8), pp 1485 - 1511
[3] Wei, J., Li, Y., Zhang, T., et al.: 'Multi - stage dynamic Bayesian network for user shopping phase modeling,' Decision Support Systems, 2024, 178, 114032
[4] Bi, X., Yang, M. C., Adomavicius, G., et al.: 'Consumer acquisition for recommender systems: A theoretical framework and empirical evaluations,' Information Systems Research, 2024, 35, (1), pp. 101–150.
[5] Adamopoulos, P., Ghose, A., Tuzhilin, A., et al.: 'Heterogeneous demand effects of recommendation strategies in a mobile application: Evidence from econometric models and machine - learning instruments,' MIS Quarterly, 2022, 46, (1), pp. 101–150
[6] Farashah, M. V., Etebarian, A., Azmi, R., et al.: 'A hybrid recommender system based - on link prediction for movie baskets analysis,' Journal of Big Data, 2021, 8, (1), pp. 32
[7] Farahani, M. G., Torkestani, J. A., Rahmani, M., et al.: 'Dynamic user profile for adaptive personalized recommender system using learning automata,' Multimedia Tools and Applications, 2023, pp. 1–21
[8] Chen, X., Yao, L., McAuley, J., et al.: 'Deep reinforcement learning in recommender systems: A survey and new perspectives', Knowledge - Based Systems,2023, 264, pp 1 - 19
[9] Fu, M., Huang, L., Rao, A., et al.: 'A Deep Reinforcement Learning Recommender System With Multiple Policies for Recommendations', IEEE Transactions on Industrial Informatics,2023, 19(2), pp 2049 - 2061
[10] Kokkodis, M., & Ipeirotis, P. G.: 'Demand-aware career path recommendations: A reinforcement learning approach', Management Science, 2021, 67(7), pp 4362-4383

[11] Bukhari, M., Maqsood, M., Adil, F., et al.: 'An actor - critic based recommender system with context - aware user modeling,' Artificial Intelligence Review, 2025, 58, (5), pp. 138–160
[12] Pang, G. Y., Wang, X. M., Wang, L., et al.: 'Efficient deep reinforcement learning - enabled recommendation with hierarchical attention and sample - enhanced priority experience replay,' IEEE Transactions on Network Science and Engineering, 2023, 10, (2), pp. 871–886
[13] Ambikesh, G., Rao, S. S., Chandrasekaran, K., et al.: 'A grasshopper optimization algorithm - based movie recommender system,' Multimedia Tools and Applications, 2023, pp. 1–22
[14] Yadav, V., Shukla, R., Tripathi, A., et al.: 'A New Approach for Movie Recommender System using K - means Clustering and PCA,' Journal of Scientific & Industrial Research, 2021, 80, (2), pp. 159–165
[15] Sahu, S., Kumar, R., Pathan, M. S., et al.: 'Movie Popularity and Target Audience Prediction Using the Content - Based Recommender System', IEEE Access, 2022,10, pp 42030 - 42046
[16] Huang, G. J., Zhu, X. T., Wasti, S. H., et al.: 'Multi - knowledge resources - based semantic similarity models with application for movie recommender system,' Artificial Intelligence Review, 2023, 56, (SUPPL 2), pp. 2151–2182