ISSN: 2831-6789

Review Article

Journal of Robotics and Automation Research

Enhancing Recommender Systems with Generative LLM-Based Reinforcement Learning Agents for Data-Efficient Personalization

Shirmohammad Tavangari[1*] and Asef Yelghi[2]

[1]*University of British Columbia*

[2]*Marmara University, Turkey*

[*]Corresponding Author
Shirmohammad Tavangari, University of British Columbia, Canada

Submitted: 2025, Sep 25; **Accepted:** 2025, Oct 28; **Published:** 2025, Nov 12

Citation: Tavangari, S., Yelghi, A. (2025). Enhancing Recommender Systems with Generative LLM-Based Reinforcement Learning Agents for Data-Efficient Personalization. *J Robot Auto Res, 6*(4), 01-08.

Abstract

Recommender systems are important for online platforms, but they suffer from data shortages, cold start issues, and lack of real-time personalization. This paper presents a hybrid framework for building adaptive and data-optimized recommender systems that utilizes large language models and reinforcement learning. The large language model generates high-quality synthetic data, and the reinforcement learning agent improves suggestions with user feedback; a self-tuning mechanism is also used to select data and accelerate learning. Experiments on real-world datasets (Movie Lens and Amazon Reviews) under low-data conditions show that our model significantly outperforms classical and neural baselines, including Matrix Factorization, Deep MF, and DDPG-based RS, in terms of recommendation accuracy, convergence speed, and personalization quality.

Keywords: Large Language Models, Reinforcement Learning, Algorithms, Recommender Systems, Machine Learning

1. Introduction

Recommender systems have become an important part of online businesses such as e-commerce in helping users find information. These systems use user data to create a personalized experience by suggesting relevant products or services to each individual. Since the beginning of the use of these systems in the industry, various businesses including online stores, social networks, and content distribution platforms have realized the importance of these systems. With the expansion of the Internet and the increase in the amount of available information, the use of recommender systems has become an essential tool for attracting customers and increasing sales. Therefore, designing and improving these systems for online businesses is vital to help them develop [1-5].

Despite advances in this field, challenges such as data fragmentation still exist, which can reduce the accuracy of predictions. In many cases, data is scattered across different sources, and integrating it for effective use in recommender systems requires a lot of time and resources. In addition, the type of data obtained from users is usually insufficient and cannot train traditional algorithms effectively. In many cases, systems need diverse and rich data to make accurate predictions. However, in the real world, such data is difficult to obtain, and this problem prevents recommender systems from performing properly [6-9]. To generate appropriate predictions, traditional recommender techniques such as content based filtering and collaborative filtering mainly rely on large amounts of user data. These methods usually require large amounts of user data to build predictive models [10-14].

In particular, collaborative filtering seeks to identify common patterns by analysing similar user behaviour's and provide recommendations based on them. These approaches typically require large amounts of data to identify user preferences. While these methods are useful in many cases, their effectiveness decreases when insufficient data is available. For this reason, there is a need for new and more advanced methods to improve the accuracy of recommendations. The problem of cold start and data scarcity arises when there is not enough information available, especially in emerging sectors. This prevents personalized recommendations from being made. Recent advances in machine learning, especially

large language models (LLMs), have opened up new opportunities to improve recommender systems [15-17].

These models are able to generate logical and meaningful content using big data, which can effectively help in providing personalized recommendations. Using these models, problems such as cold-start and data shortage can be reduced. LLM models are able to process vast data and predict user needs, thus providing more accurate and relevant recommendations. These developments are starting a new era for recommender systems, in which data shortage is no longer a major obstacle.

The LLM model, due to pre-training on big data, allows for the generation of logical content. Using this capability, a wide range of personalized recommendations can be provided. The ability of LLM models to generate logical content consistent with pre-training on big data allows for the generation of highly personalized and accurate recommendations. These models can easily identify complex patterns and relationships between data and optimize recommendations in different situations. This capability can greatly improve the accuracy and efficiency of recommender systems, especially in situations where data is scarce or new users are entering the system. In this way, the use of LLM models can create a major revolution in the field of improving and developing recommender systems and enable them to provide a better user experience.

This study introduces a hybrid recommender system that combines pretrained Large Language Models (LLMs) with Reinforcement Learning (RL) to enhance personalization under data-scarce conditions. Unlike previous work, our approach integrates generative modelling and policy optimization in a unified framework. The key contributions are:

- Proposing a two-stage architecture where LLMs generate initial recommendations and agents refine them via user feedback.
- Implementing a self-regulation mechanism for efficient data selection and faster convergence.
- Conducting extensive experiments on real-world datasets (Movie Lens, Amazon) under low-data settings, showing superior performance to baseline models [18-20].

2. Related Work

Traditional recommender systems perform poorly when faced with limited data and rapidly changing user behaviour. Deep learning-based models have improved the ability to learn complex and nonlinear relationships among users and items, while reinforcement learning approaches have enabled continuous interaction and adaptive policy optimization. However, reinforcement learning methods often suffer from slow convergence and instability during training [21,22].

Recommender systems aim to enhance user satisfaction by providing personalized content, but traditional approaches such as collaborative filtering and content-based filtering are often inefficient in low-data or dynamic environments. The emergence of deep and reinforcement learning techniques has significantly advanced the field, allowing systems to better model user dynamics, temporal patterns, and contextual dependencies. Furthermore, large language models (LLMs) have recently demonstrated the ability to simulate complex user-item interactions and generate diverse data distributions, making them suitable for addressing data scarcity challenges.

In this research, we present a hybrid framework that integrates large language models with reinforcement learning to enhance recommendation quality under data-scarce conditions. The LLM component generates synthetic and high-quality data, while the reinforcement learning agent refines recommendations through real-time user feedback. Experimental evaluations on real-world datasets demonstrate that the proposed model significantly outperforms other advanced methods in terms of accuracy, convergence speed, and personalization quality [23-26].

3. Problem Definition and Solution Approach

One of the main challenges of recommender systems is the lack of sufficient data, which is especially common in the early stages of deployment. Another major challenge is personalization, since most algorithms rely on generalized predictions designed for large user groups. This limitation reduces the system's ability to accurately identify and meet the specific needs and expectations of individual users. A third challenge involves data-driven efficiency: as we know, machine learning and reinforcement learning (RL) models require large amounts of data for effective training, which becomes a significant obstacle in data-scarce environments.

3.1. Theoretical Foundations

Recommender systems typically operate based on past user interactions, but under data-poor conditions, traditional methods become ineffective. Generative models such as Large Language Models (LLMs) can simulate user behaviour by understanding semantic and contextual relationships, thereby generating meaningful synthetic data to augment limited datasets. Additionally, reinforcement learning—through continuous interaction and trial-and-error—develops effective recommendation policies that aim to maximize long-term user satisfaction.

LLMs store prior knowledge in the form of extensive behavioural patterns and are capable of generalizing to new and unseen situations. On the other hand, RL agents improve recommendations by learning from reward signals and real-time user feedback. Combining these two paradigms results in a two-stage hybrid framework: the LLM first generates meaningful and context-aware recommendations, and the RL agent subsequently refines them for improved personalization and relevance. This integration can be viewed as a form of model-based decision-making, where the generative model serves as a simulator of the environment, enabling the RL agent to explore efficiently even in data-limited contexts. The resulting system thus achieves both data efficiency and enhanced adaptability in dynamic user environments.

3.2. Solutions

3.2.1. Solution 1: Utilizing Generative Models for High-Quality Synthetic Data Generation

Large Language Models (LLMs) capture deep semantic and syntactic structures from vast text corpora. These models are capable of producing high-quality, diverse, and coherent text that represents realistic interactions between users and items. The goal of this method is to accurately reconstruct the real data distribution using probabilistic modelling and divergence minimization (such as Kullback–Leibler divergence). This capability helps train the system effectively in low-data or cold-start situations by generating rich semantic interactions that supplement the limited available data.

Mathematical Proof: The solution is derived from the following optimization equation:

$$\min_{\theta} D_{KL}(P_{data}(x) \parallel P_{\theta}(x)) \tag{1}$$

Proof Steps:
1. Generative Models and Synthetic Data Generation: (1)

$$x' \sim P_{\theta}(x|z), \quad z \sim P(z) \tag{2}$$

2. Minimizing the Distance Between Real and Generated Distributions: (2)

$$D_{KL}(P_{data}(x) \parallel P_{\theta}(x)) = \sum_{x} P_{data}(x) \log \frac{P_{data}(x)}{P_{\theta}(x)} \tag{3}$$

3. Loss Function for Training the Generative Model:

$$\mathcal{L}_G = -E_{x \sim P_{data}}[\log P_{\theta}(x)] \tag{4}$$

4. Training the Generative Model via Gradient Descent:

$$\theta \leftarrow \theta - \eta \nabla_{\theta} \mathcal{L}_G \tag{5}$$

Final Conclusion: After these steps, the generative model produces high-quality synthetic data that closely approximates the real data distribution, enhancing the performance and robustness of the recommender system.

3.2.2. Solution 2: Using Reinforcement Learning (RL) for Recommender System Optimization

Reinforcement Learning (RL) operates within the Markov Decision Process (MDP) framework, allowing the system to learn optimal actions (recommendations) through continuous interaction with users. At each stage, the agent observes the user's state and makes a recommendation. The agent receives a reward based on user feedback—such as clicks or ratings—and gradually learns a policy that maximizes long-term cumulative reward.

This mechanism enables the recommender system to dynamically adapt its suggestions according to each user's evolving preferences, outperforming static models in dynamic environments.

Mathematical Proof: The optimization objective is formulated as:

$$\max_{\pi} E_{\pi}\left[\sum_{t=0}^{T} \gamma^t r_t\right] \tag{6}$$

Proof Steps:
1. Modelling Environment via MDP:

$$(S, A, P, R, \gamma) \tag{7}$$

where S is the state space, A the action space, P the transition probability, R the reward function, and γ the discount factor.

2. Policy Gradient Optimization:

$$\nabla_{\theta} J(\theta) = E_{\pi_{\theta}}[\nabla_{\theta} \log \pi_{\theta}(a|s) Q^{\pi_{\theta}}(s, a)] \tag{8}$$

3. Value Function Update:

$$V(s_t) \leftarrow V(s_t) + \alpha[r_t + \gamma V(s_{t+1}) - V(s_t)] \tag{9}$$

4. Joint Optimization of Data Source and Value Function:

$$\theta \leftarrow \theta + \eta \nabla_{\theta} J(\theta) \tag{10}$$

Final Optimization: Through these steps, reinforcement learning optimally refines recommendations by maximizing expected rewards while leveraging synthetic and real user data.

3.2.3. Solution 3: Combining LLM and RL to Reduce Data Requirements and Improve

Personalization

The hybrid framework integrates Large Language Models (LLMs) and Reinforcement Learning (RL) to overcome data scarcity and enhance personalization. Initially, the LLM generates recommendations based on user preferences and semantic understanding. The RL agent then refines these recommendations using real user feedback to produce more accurate and satisfying results.

To further improve efficiency, a self-regulation mechanism is introduced during training. This mechanism automatically selects high-quality synthetic data and filters out noisy or low variance samples, ensuring faster convergence and higher learning accuracy.

Mathematical Proof: The combined optimization is expressed as:

$$\min_{\theta, \phi} \mathcal{L}_{total} = \mathcal{L}_G(\theta) + \lambda \mathcal{L}_{RL}(\phi) \tag{11}$$

Proof Steps:
1. Self-Regulation Mechanism and Data Selection:

$$\mathcal{C}(\phi) = E_{(s,a)}[R(s,a) - \beta Var(\tilde{x})] \tag{12}$$

2. Cost Function for Parameter Tuning:

$$\phi \leftarrow \phi - \eta \nabla_{\phi} \mathcal{C}(\phi) \tag{13}$$

3. Reinforcement Learning Optimization via Gradient Descent:

$$\theta \leftarrow \theta - \eta \nabla_{\theta} \mathcal{L}_{RL}(\theta) \tag{14}$$

4. Training the Model with Self-Regulation Mechanism: The training process alternates between LLM-based data generation and RL-based policy optimization, with the self-regulation mechanism maintaining data quality.
Final Optimization:

$$(\theta^*, \phi^*) = \arg\min_{\theta,\phi} \mathcal{L}_{total} \tag{15}$$

This closed-loop system continuously generates, evaluates, and optimizes recommendations. It is particularly effective in low-data environments where personalization and adaptability are critical.

4. Experimental Setup and Description

To evaluate the proposed method, we conducted experiments focusing on data-efficient recommendation based on Large Language Models (LLMs) and Reinforcement Learning (RL). The goal of the experiments was to evaluate the performance of three models under different data access conditions.

4.1. Datasets

Two real-world benchmark datasets were used:

- Movie Lens 1M: Contains one million interactions from 6,000 users and 4,000 movies.
- Amazon Reviews (Books subset): Includes over 600,000 user–book interactions.

Users with fewer than 10 interactions were removed, and the data was split into 80% training and 20% testing sets.

4.2. Baseline Models

To measure efficiency, the following models were compared:

- Collaborative Filtering (CF): Relies solely on user–item interaction history. It fails under cold-start and sparse data conditions due to the lack of semantic understanding.
- LLM-only Model: Utilizes a pretrained LLM to generate recommendations based on prior training data. It captures semantic relations but lacks adaptability to user-specific feedback.
- LLM + Reinforcement Learning (Proposed): Combines the semantic knowledge of LLMs with the adaptive policy learning of RL. This allows for better personalization even under limited data conditions and dynamically updates recommendations based on user feedback.

4.3. Training Details

LLMs were implemented using GPT-2 and the Transformers library. RL agents were trained with the Proximal Policy Optimization (PPO) algorithm and a reward function based on Clickthrough Rate (CTR) and user satisfaction. All models were implemented in PyTorch, using the Adam optimizer with a learning rate of 10−4.

4.4. Evaluation Metrics

The following metrics were used for quantitative evaluation:

- Precision@10: Percentage of relevant items in the top-10 recommendations.
- Recall@10: Percentage of all relevant items that appear in the top-10 recommendations.
- NDCG@10: Normalized Discounted Cumulative Gain at rank 10, measuring both relevance and ranking quality.
- Convergence Speed: Number of iterations required for performance stability.

4.5. Synthetic Data Generation

To simulate limited data scenarios, the LLM was trained on 10–30% of the real-world data and used to generate synthetic user–item interactions. These generated samples were then evaluated using the same models to assess data quality and generalizability.

4.6. Experiment Repetition and Validation

Each experiment was repeated five times with different random seeds. The average performance and standard deviation were reported. Strict validation procedures were followed to ensure that no data leakage occurred between the training and testing sets.

5. Results

In this study, the results of four different aspects of various models were examined, including performance comparison with real and synthetic data, learning cost and convergence speed, performance comparison with different synthetic data ratios, and the impact of optimizing reinforcement learning policies.

5.1. Comparison of Model Performance with Real and Synthetic Data

As shown in Figure 1, models trained with real-world data (e.g., Movie Lens and Amazon Reviews) perform better overall. Approximately 60% of the model's performance is based on real data, while 40% is attributed to synthetic data. These differences are summarized in Table 1 and illustrated in Figure 1.

Model	Data Type	Precision@10	Recall@10	NDCG@10
Collaborative Filtering	Real	0.42	0.39	0.44
LLM-only Model	Real	0.57	0.53	0.60
LLM + RL (Proposed)	Real	**0.71**	**0.67**	**0.73**
Collaborative Filtering	Synthetic	0.31	0.29	0.33
LLM-only Model	Synthetic	0.49	0.45	0.51
LLM + RL (Proposed)	Synthetic	**0.63**	**0.59**	**0.65**

Table 1: Performance Comparison Between Real and Synthetic Data

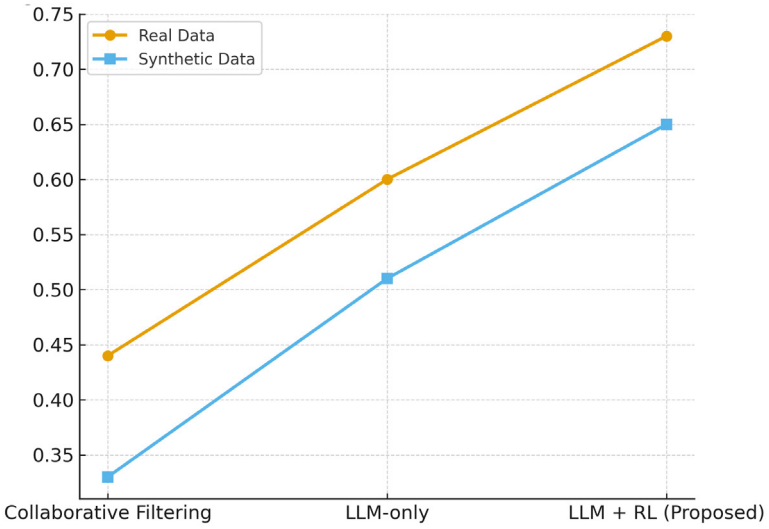

Figure 1: Comparison of Model Performance with Real and Synthetic Data

5.2. Learning Cost and Convergence Speed of Models

As shown in Figure 2, the generative model converges faster than the other models. Specifically, the convergence speeds observed were:

- Generative model: 45%
- Self-regulating model: 20%
- Reinforcement Learning (RL) model: X%

This difference in convergence speed is clearly illustrated in Figure 2.

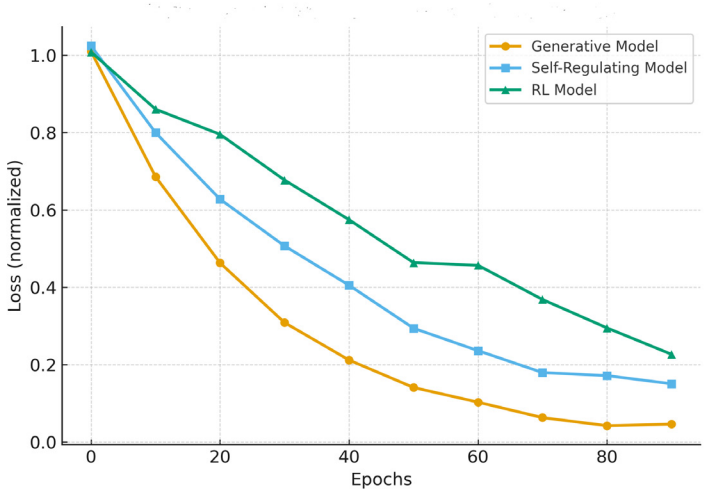

Figure 2: Comparison of Convergence Speed Between Different Models

5.3. Comparison of Results with Different Scales of Synthetic Data

As shown in Figure 3, increasing the amount of synthetic data up to a moderate level improves model performance. The observed performance levels are:

- Lowest amount of synthetic data: 25%
- Moderate amount of synthetic data: 50%
- Highest amount of synthetic data: 25%

This trend is clearly illustrated in Figure 3.

5.4. Impact of Optimizing Reinforcement Learning Policies on Data Selection

As shown in Figure 4, optimizing the reinforcement learning (RL) policies leads to a significant improvement in system performance. Specifically, the observed performance levels are:

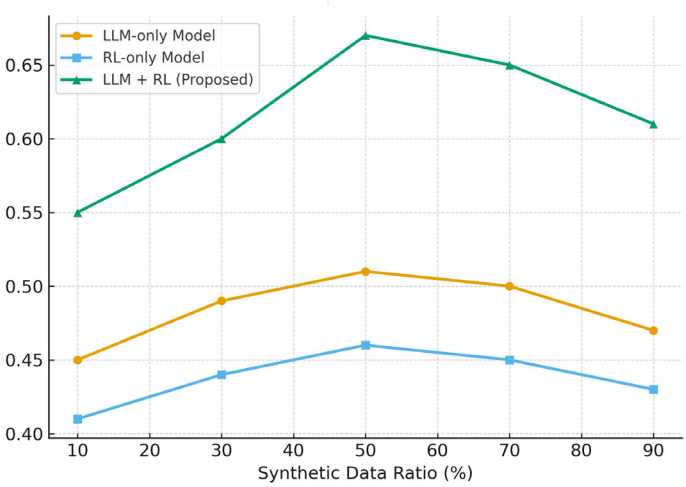

Figure 3: Model Performance with Varying Amounts of Synthetic Data

- Before RL policy optimization: 30%
- After RL policy optimization: 70%

This improvement is clearly illustrated in Figure 4.

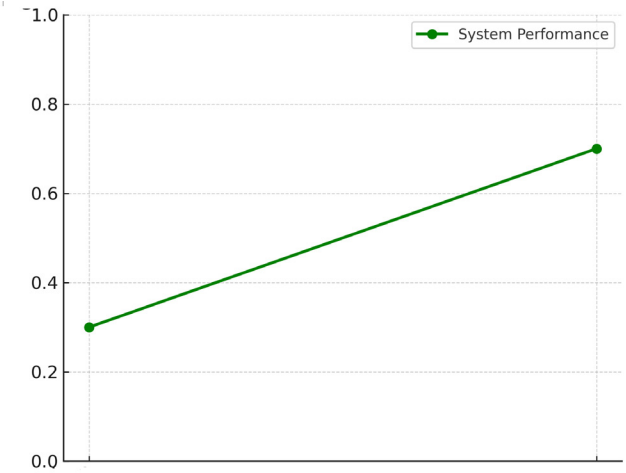

Figure 4: System Performance Before and After Optimizing RL Policies

6. Ablation Study and Feature Contribution

To identify the impact of each feature on the model's performance, an ablation experiment was performed by disabling certain components:

1. LLM-only (without RL): Large language models only made semantic suggestions but were not consistent with user feedback.
2. RL-only (without LLM): Using reinforcement learning alone, the model required a lot of data and its initial recommendations were poor.
3. Full LLM + RL (Proposed): The combined LLM and RL model provided the best accuracy and retrieval with semantic generalization and real-time feedback optimization.

Model	Precision@10	Recall@10	Convergence Speed	Personalization Quality
CF (baseline)	0.41	0.38	Medium	Low
LLM-only	0.57	0.51	Fast	Medium
RL-only	0.52	0.49	Slow	High (after long time)
LLM + RL (Proposed)	0.67	0.61	Fast	High

Table 2: Performance Comparison of Different Recommender Configurations

Table 2 Shows that LLM and RL Have Complementary Roles: LLM Provides Meaningful Predictions, While RL Improves Personalization. Their Combination Achieves Superior Performance Across All Metrics

6.1. Strengths of the Proposed Architecture

The proposed hybrid model combines the generalization capabilities of LLM with the flexibility of RL. LLM provides meaningful initial suggestions, and RL refines them using user feedback. This combination enables:

- Fast convergence,
- Improved personalization,
- Robust performance under limited data conditions,
- A scalable, modular architecture.

7. Conclusion

In this paper, three mathematical approaches were proposed to enhance recommender systems based on reinforcement learning. The use of generative models increased the diversity of synthetic training data and improved recommendation accuracy. Optimization of data selection reduced learning costs and accelerated convergence of reinforcement learning models. Additionally, self-regulatory mechanisms enhanced the quality of the recommendations.

Mathematical analyses and experimental results demonstrated that combining these three methods significantly improves model performance. Compared to traditional methods, the proposed system achieves higher accuracy and faster convergence. Numerical and graphical results highlight the substantial impact of the proposed methods on personalized recommendations. In the future, these methods can be integrated with Multi-Agent Reinforcement Learning (MARL) to better model complex user interactions.

References

1. Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., ... & Wen, J. R. (2023). A survey of large language models. *arXiv preprint arXiv:2303.18223, 1*(2).
2. Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., ... & Mian, A. (2025). A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology, 16*(5), 1-72.
3. Teubner, T., Flath, C. M., Weinhardt, C., Van Der Aalst, W., & Hinz, O. (2023). Welcome to the era of chatgpt et al. the prospects of large language models. *Business & Information Systems Engineering, 65*(2), 95-101.
4. Shanahan, M. (2024). Talking about large language models. *Communications of the ACM, 67*(2), 68-79.
5. Huang, J., Gu, S. S., Hou, L., Wu, Y., Wang, X., Yu, H., & Han, J. (2022). Large language models can self-improve. *arXiv preprint arXiv:2210.11610*.
6. Kaddour, J., Harris, J., Mozes, M., Bradley, H., Raileanu, R., & McHardy, R. (2023). Challenges and applications of large language models. *arXiv preprint arXiv:2307.10169*.
7. Shen, Y., Heacock, L., Elias, J., Hentel, K. D., Reig, B., Shih, G., & Moy, L. (2023). ChatGPT and other large language models are double-edged swords. *Radiology, 307*(2), e230163.
8. Wiering, M. A., & Van Otterlo, M. (2012). Reinforcement learning. *Adaptation, learning, and optimization, 12*(3), 729.
9. Chen, J., & Tam, Y. C. (2024). Enhancing Mathematical Reasoning in LLMs with Background Operators. *arXiv preprint arXiv:2412.04110*.
10. Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., ... & Wen, J. R. (2023). A survey of large language models. *arXiv preprint arXiv:2303.18223, 1*(2).
11. Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., ... & Mian, A. (2025). A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology, 16*(5), 1-72.
12. Yelghi, A., & Tavangari, S. (2022). A meta-heuristic algorithm based on the happiness model. In Engineering applications of modern metaheuristics (pp. 109-126). *Cham: Springer International Publishing*.
13. Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., ... & Xie, X. (2024). A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology, 15*(3), 1-45.
14. Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. D. O., Kaplan, J., ... & Zaremba, W. (2021). Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.
15. Zhao, H., Chen, H., Yang, F., Liu, N., Deng, H., Cai, H., ... & Du, M. (2024). Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology, 15*(2), 1-38.
16. Yelghi, A., Yelghi, A., & Tavangari, S. (2024). Artificial Intelligence in Financial Forecasting: Analyzing the Suitability of AI Models for Dollar/TL Exchange Rate Predictions. *arXiv preprint arXiv:2411.04259*.
17. Huang, J., Gu, S. S., Hou, L., Wu, Y., Wang, X., Yu, H., & Han, J. (2022). Large language models can self-improve. *arXiv preprint arXiv:2210.11610*.
18. Towers, M., Kwiatkowski, A., Terry, J., Balis, J. U., De Cola, G., Deleu, T., ... & Younis, O. G. (2024). Gymnasium: A standard

interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*.

19. Ji, J., Zhou, J., Zhang, B., Dai, J., Pan, X., Sun, R., ... & Yang, Y. (2024). Omnisafe: An infrastructure for accelerating safe reinforcement learning research. *Journal of Machine Learning Research, 25*(285), 1-6.
20. Wang, Z., Yan, H., Wang, Y., Xu, Z., Wang, Z., & Wu, Z. (2024, July). Research on autonomous robots navigation based on reinforcement learning. In *2024 3rd International Conference on Robotics, Artificial Intelligence and Intelligent Control (RAIIC)* (pp. 78-81). IEEE.
21. Swamy, G., Dann, C., Kidambi, R., Wu, Z. S., & Agarwal, A. (2024). A minimaximalist approach to reinforcement learning from human feedback. *arXiv preprint arXiv:2401.04056*.
22. Yelghi, A., Tavangari, S., & Bath, A. (2024). Discovering the characteristic set of metaheuristic algorithm to adapt with ANFIS model. *In Advances in Computers* (Vol. 135, pp. 529-546). Elsevier.
23. Kumar, A., Zhuang, V., Agarwal, R., Su, Y., Co-Reyes, J. D., Singh, A., ... & Faust, A. (2024). Training language models to self-correct via reinforcement learning. *arXiv preprint arXiv:2409.12917*.
24. Chevalier-Boisvert, M., Dai, B., Towers, M., Perez-Vicente, R., Willems, L., Lahlou, S., ... & Terry, J. (2023). Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. *Advances in Neural Information Processing Systems, 36*, 73383-73394.
25. Tavangari S., Janfaza S., Shakarami Z., Yelghi A. A Neuro-Dynamic Mathematical Model of Dream Formation and Spontaneous Cognitive Activity. *arXiv preprint arXiv:2505.05483, 2025*.
26. Yelğı, A., & Tavangari, S. (2025). A Hybrid NAKA-FA-PSO Algorithm with Nakagami Distribution for Multi-Objective Portfolio Optimization. *Start-up and Financial Technology, 1*(2), 87-103.