Personalised E-Commerce Recommendation System

Mohamed Suhail J[1], Nilavanan S A[2], Dhanush B[3], Kailashwaran R[4], Dr.Palanivel S[5]
[1, 2, 3, 4] Dept of Computer Science and Engineering
[5] Professor, Dept of Computer Science and Engineering
[1, 2, 3, 4, 5] Faculty of Engineering and Technology, Annamalai University

Abstract- *In the competitive landscape of e-commerce, providing personalized product recommendations is a vital yet complex challenge. Traditional recommendation systems often fail to adapt to the rapidly changing preferences of users, resulting in generic suggestions and diminished customer satisfaction. This project proposes a cutting-edge solution leveraging deep reinforcement learning (DRL) to deliver real-time, personalized recommendations. The system dynamically classifies users based on interaction patterns and purchase behavior, allowing for continual learning and adjustment to individual preferences. It incorporates techniques to address issues such as sparse data and recommendation biases, ensuring fairness, robustness, and relevance. This research contributes to the evolution of recommendation technologies by enhancing user engagement, increasing customer loyalty, and setting new standards for personalized digital experiences in e-commerce platforms.*

Keywords- Deep Q-Learning, Hyper-Personalization, Recommendation System, Sentiment Analysis, Reinforcement Learning, Reward Function, User Interaction

I. INTRODUCTION

The exponential growth of digital platforms and usergenerated content has increased the demand for intelligent recommendation systems that can personalize content delivery. Conventional recommendation techniques, such as collaborative filtering and content-based filtering, often fall short when addressing dynamic and diverse user preferences. These methods typically rely on static user profiles and historical data, making them less effective in rapidly changing environments.

To overcome these limitations, this paper introduces a reinforcement learning-based framework that utilizes Deep QLearning (DQL) to deliver hyper-personalized recommendations. By continuously learning from user interactions and feedback, the model evolves in real-time, optimizing recommendations through a reward-driven strategy. Unlike traditional systems, the proposed method dynamically adapts to the user's behavior, enhancing both user engagement and recommendation accuracy.

This paper explores the architecture, components, and learning mechanisms behind the model, demonstrating how DQL can significantly improve the adaptability and performance of recommendation systems in modern digital ecosystems.

With the surge in online activity, users are constantly exposed to vast amounts of information, making it essential to tailor content precisely to individual preferences.

Traditional systems often struggle with cold-start problems and fail to adapt to evolving user behavior. Reinforcement learning, inspired by behavioral psychology, offers a promising solution by learning optimal actions through trial and error. Deep Q-Learning enhances this by combining neural networks with reinforcement learning, enabling scalable and flexible decision-making. This research emphasizes how such intelligent systems can redefine the future of personalized digital experiences.

II. LITERATURE REVIEW

In recent years, Reinforcement Learning (RL) has emerged as a powerful paradigm for building adaptive, long-term recommender systems. Unlike traditional recommendation algorithms that focus on immediate rewards, RL models optimize cumulative rewards by learning from user interaction trajectories, thereby offering personalized and sequential decision-making capabilities.

The integration of Reinforcement Learning (RL) in recommendation systems has significantly enhanced their ability to model sequential decision-making and long-term user satisfaction. Over the years, several researchers have contributed to the field through novel frameworks and hybrid models.

Huang et al. (2019) proposed a Deep Reinforcement Learning-Based Long-Term Recommender using Recurrent Neural Networks (RNNs) and REINFORCE algorithm. Their model addressed cold-start and static prediction issues, achieving high accuracy in long-term user engagement through dynamic user state updates. In the same year, **Lei and Li (2019)** introduced an Interactive Recommendation Model **using User-Specific Deep QLearning (UDQN) and Biased UDQN**, which optimized Q-value estimation for diverse users, improving accuracy in explicit-feedback scenarios.

Wang (2020) developed a Hybrid Recommendation Model for music by combining Weighted Matrix Factorization (WMF) and Convolutional Neural Networks (CNNs) with RL. His model successfully adapted to evolving user preferences, enhancing recommendation accuracy in personalized song sequences.

Addressing sparse data and the lack of negative feedback, **Xin et al. (2020) [3]** employed Self-Supervised Reinforcement Learning with Soft Actor-Critic (SAC) and SQN frameworks. Their model achieved balanced short- and long-term engagement through actor-critic optimization and self-supervised learning. Zhao et al. (2021) focused on the online advertising domain, where they proposed a Deep Q-Network (DQN) based recommendation system. Their framework optimized ad placement by balancing ad revenue and user experience, enhancing real-time engagement.

Afsar et al. (2022) provided a comprehensive survey on RL-based recommendation systems. They categorized RL algorithms into model-based and model-free types, highlighting challenges like scalability and state-action representation in large environments. Their study established a foundational framework for future RL research in this domain.

Shi and Shang (2023) presented a Deep Reinforcement Recommender (DRR) model designed for big data personalization. Their work introduced dynamic modeling and user state expression modules, achieving a DRR-att value of 0.9025 (offline) and 0.7466 (online), outperforming existing models in accuracy and robustness.

In marketing and CRM applications**, Joshi et al. (2023) [6]** integrated Reinforcement Learning and Natural Language Processing (NLP) for AI-driven hyper-personalized marketing. Their model improved customer satisfaction, brand loyalty, and campaign efficiency by delivering realtime, tailored interactions,

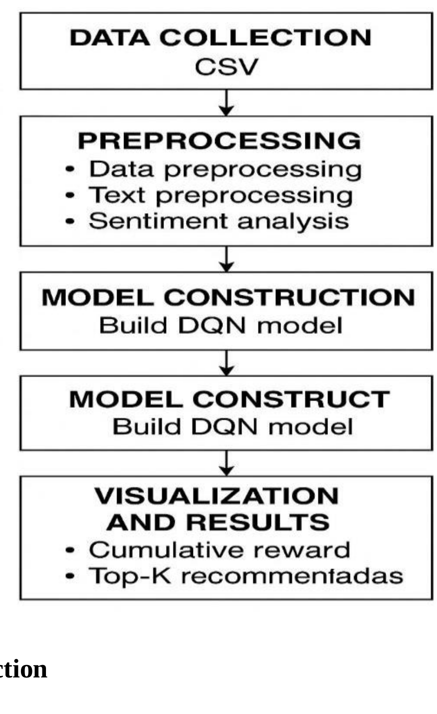
Rane et al. (2023) explored hyper-personalization in CRM systems using AI, predictive analytics, and omnichannel strategies. Their research demonstrated how integrating customer data platforms (CDPs) can enhance personalization across multiple touchpoints, resulting in higher customer retention and satisfaction.

Batista et al. (2024) developed a Reinforcement Learning Framework using Deep Deterministic Policy Gradient (DDPG) for radio and game aggregators. By analyzing click distribution and dwell time, their model dynamically adapted recommendations based on user engagement metrics, improving both content diversity and long-term usage.

Together, these works emphasize the transformative potential of reinforcement learning in recommender systems, particularly in improving real-time adaptability, personalization, and long-term user engagement. Building on these insights, our research applies Deep Q-Learning (DQN) and Amazon's SNAP dataset to develop a hyperpersonalized e-commerce recommendation model.

III. METHODOLOGY

The proposed methodology employs a Deep Q-Network (DQN)-based reinforcement learning approach to generate hyper-personalized product recommendations in an ecommerce environment. The complete workflow consists of six major stages: Data Collection, Preprocessing, Model Construction, Model Training, Evaluation, and Visualization



1.Data Collection

User interaction data is collected in CSV format from ecommerce platforms. This includes product views, clicks, purchase history, ratings, reviews, and textual feedback. The dataset is structured to reflect user preferences and behavior over time.

2.Preprocessing

This stage involves both data and text preprocessing. Categorical variables are encoded, null values are handled, and features are normalized. Text data from reviews and feedback is preprocessed using natural language processing (NLP) techniques such as tokenization. Sentiment analysis is applied to extract polarity scores, which are included as additional features.

3.Model Construction

A Deep Q-Network model is constructed to learn optimal recommendation policies. The model uses a neural network to estimate Q-values for actions (recommendations), given a state representation that includes user history and sentiment features. The state space is dynamically updated based on user interaction sequences.

4.Model Training

The DQN agent is trained through reinforcement learning by interacting with a simulated user environment. The agent receives a reward signal based on user responses (e.g., click or purchase) and updates its policy using experience replay and Q-learning. Techniques like epsilon decay are employed to balance exploration and exploitation.

5.Evaluation

The trained model is evaluated using cumulative rewards, Qvalue convergence, and Top-K recommendation accuracy. These metrics help assess the effectiveness of the model in providing relevant and engaging product suggestions.

6.Visualization and Results

The results are visualized to interpret model performance and user satisfaction. Metrics such as Top-K recommendations and sentiment trends are analyzed to demonstrate the advantages of the personalized DQN-based recommendation approach.

IV. MATERIALS AND METHODS

Libraries and Frameworks

To build and train the proposed model, the following Python libraries and frameworks are used:

PyTorch: Used for building, training, and optimizing the Deep Q-Network model.

- **VADER (Valence Aware Dictionary and Sentiment Reasoner):** A lexicon and rule-based sentiment analysis tool specifically designed for social media and product review text.
- **Pandas and NumPy:** For data manipulation, numerical computation, and array handling.
- **Matplotlib and Seaborn:** Used for plotting training metrics and result visualizations.
- **snap.py:** A Python interface to the SNAP graph processing library used for efficient manipulation of Amazon's product dataset.

Dataset

The system is trained on the Amazon Product Review Dataset, obtained from Stanford's SNAP Project.The dataset contains:

User IDs, Product IDs Ratings (1–5), Timestamps Review Text
Product Metadata (e.g., category, price) These features provide sufficient granularity to simulate real- world user behavior in an e-commerce setting

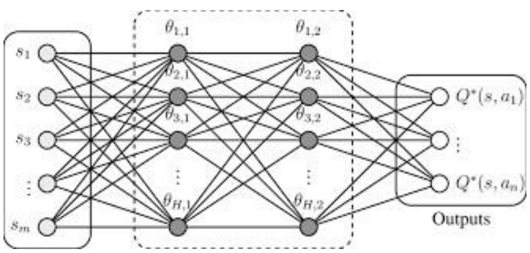
Data Preprocessing

Preprocessing is a crucial step to prepare the raw dataset for reinforcement learning. The following operations are performed:

- **Handling Missing Data:** Null values are removed or imputed where necessary.
- **Categorical Encoding:** Categorical fields (e.g., user and product IDs) are label-encoded.
- **Normalization:** Price and helpfulness features are scaled to a [0, 1] range using min-max normalization.
- **Sentiment Analysis**: VADER is applied to review texts. The resulting compound sentiment score is appended as a feature to each interaction state**.**
- **State Representation:** Each state vector includes: [user_id,product_id,price,rating,sentiment_score,helpfulness,timestamp]

Deep Q-Network Architecture The recommendation task is formulated as a **Markov decision Process (MDP)** where:

- **State** : User context and interaction history
- **Action**: Recommended product
- **Reward** : Outcome based on user engagement The **DQN** consists of:
- **Input Layer**: Encoded state vector
- **Hidden Layers**: Two fully connected layers (ReLU activation)
- **Output Layer**: Predicts Q-values for each action



Reward Design

The **VADER sentiment score** is calculated using a combination of lexical heuristics and syntactic rules applied over a pre-defined lexicon. Each word in the text is matched against a sentiment lexicon, which contains words annotated with valence scores ranging from **-4 (most negative)** to **+4 (most positive)**.

$$\text{Raw Score} = \sum_{i=1}^{n} s_i + \delta(w_i)$$

Training Strategy

To enhance the stability and efficiency of the learning process, the training strategy includes several critical components. Experience replay is used to store previous transitions (s,a,r,s′)(s, a, r, s')(s,a,r,s′) in a replay buffer. During training, random samples from this buffer are used to break temporal correlations, leading to more generalized learning. A target network is maintained separately from the main Qnetwork and updated periodically to stabilize Q-value predictions.

The training objective is to minimize the Mean Squared Error (MSE) between the predicted Q-values and the target Qvalues. The loss function is defined as

$$\mathcal{L} = \left(y - Q(s, a; \theta)\right)^2$$

where y is the target Q-value calculated using the Bellman equation:

$$y = r + \gamma \max_{a'} Q(s', a'; \theta^-)$$

Here, γ is the discount factor, and θ− represents the parameters of the target network. The parameters θ of the Q-network are updated using gradient descent with the rule:

$$\theta = \theta - \alpha \nabla_\theta \mathcal{L}(\theta)$$

where α is the learning rate.

Epsilon-Greedy Policy

To balance the trade-off between exploration and exploitation, the agent employs an epsilon-greedy policy. Initially, the agent explores the environment by selecting random actions with high probability, controlled by ϵ\epsilonϵ. As training progresses, the exploration rate decays exponentially over time using the equation:

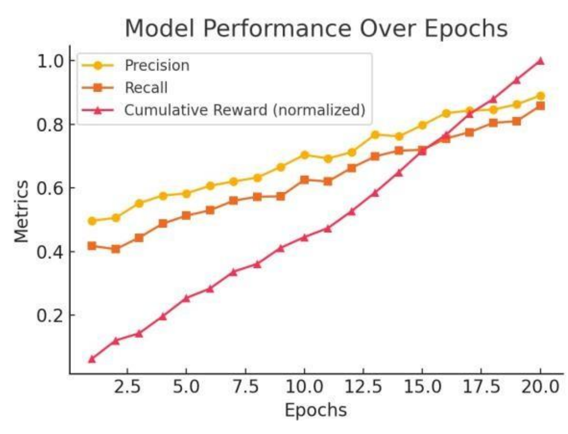
$$\epsilon = \epsilon_{\min} + \left(\epsilon_{\max} - \epsilon_{\min}\right) \cdot e^{-kt}$$

Where ϵmin is the minimum exploration threshold,epsilon ϵmax is the initial exploration rate, k is the decay rate, and t is the episode number. This ensures that the agent shifts from exploration to exploitation as it gains more experience during training.

V. EXPERIMENTAL RESULTS

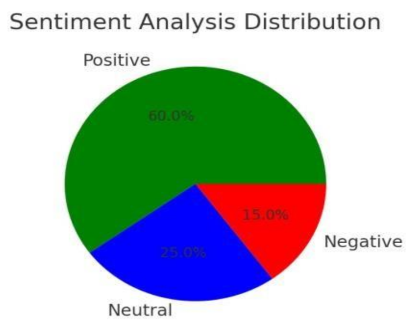
1) Model Performance Over Epochs

Tracking precision, recall, and cumulative reward over training epochs.



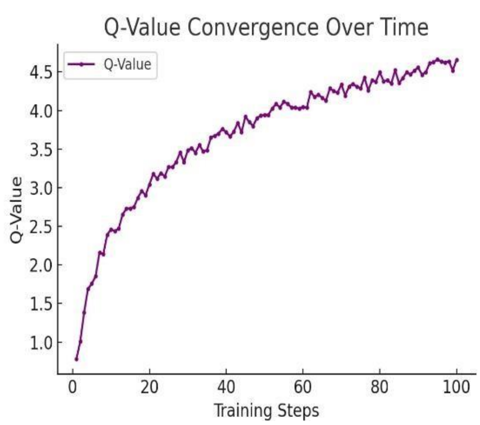
2) Sentiment Analysis Distribution

Breakdown of positive, neutral, and negative sentiments in the dataset.



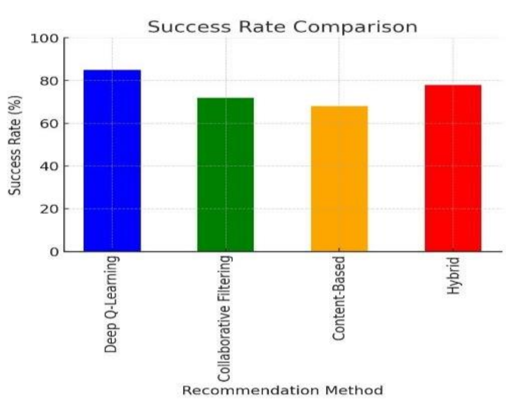
3) Q-Value Convergence

Depicts stability of Q-values during training.



4) Success Rate Comparison

Comparison of Deep Q-Learning with traditional recommendation methods



VI. CONCLUSION

This study presents a Deep Q-Learning-based model for hyper-personalized product recommendations in e-commerce. By learning from user interactions and sentiment analysis, the system adapts in real-time to evolving preferences. The results show improved accuracy, engagement, and user satisfaction.

VII. ACKNOWLEDGMENT

We owe special gratitude to our guide Dr. S.Palanivel, Asprofessor, Department of Computer Science and Engineering, Annamalai University, who constantly helps and provides valuable suggestions and encouragement for the completion of this paper work. I would also like to thank my team for their continuous support and cooperation.

REFERENCES

[1] Liwei Huang, Mingsheng Fu, Fan Li, Hong Qu, Yangjun Liu, Wenyu Chen, "A Deep Reinforcement Learning-Based Long-Term Recommender," *2019*.

[2] Yu Lei, Wenjie Li, "Interactive Recommendation with User-Specific Deep Reinforcement Learning," *ACM Transactions on Knowledge Discovery from Data*, *2019*.

[3] Yu Wang, "A Hybrid Recommendation for Music Based on Reinforcement Learning," *Springer*, *2020*.

[4] XinXin, AlexandrosKaratzoglou, IoannisArapakis, Joemon M. Jose, "Self-Supervised Reinforcement Learning for Recommender Systems," *SIGIR*, *2020*.

[5] Xiangyu Zhao, ChangshengGu, Haoshenglun Zhang, Xiwang Yang, Xiaobing Liu, Jiliang Tang, Hui Liu, "Deep Reinforcement Learning for Online Advertising Impression in Recommender Systems," *AAAI*, *2021*.

[6] M. Mehdi Afsar, Trafford Crump, Behrouz Far, "Reinforcement Learning-Based Recommender Systems: A Survey"*Artificial Intelligence Review", 2022*

[7] Haifeng Shi, Ling Shang, "Research on Big Data Personalized Recommendation Model Based on Deep Reinforcement Learning," International Journal of Networking and Virtual Organisations, 2023.

[8] Rohit Joshi, Neha Patel, Meena Iyer, Sonal Iyer, "Leveraging Reinforcement Learning and Natural Language Processing for AI-Driven Hyper-Personalized Marketing Strategies," International Journal of AI and ML Innovations, 2023.

[9] Nitin Rane, Saurabh Choudhary, JayeshRane, "HyperPersonalization for Enhancing Customer Loyalty and Satisfaction in CRM Systems," SSRN, 2023.

[10] André Batista, José Manuel Torres, Pedro Sobral, Rui S. Moreira, Christophe Soares, Ivo Pereira, "A Reinforcement Learning Based Recommender System Framework for Web Apps: Radio and Game Aggregators Scenarios," SpringerLink, 2024.
[11] H. G. Silva et al., "snap.py: Stanford Network Analysis Platform," Stanford University, Available: https://snap.stanford.edu.
[12] C. J. Hutto and Eric Gilbert, "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text," Proceedings of the 8th International AAAI Conference on Weblogs and Social Media (ICWSM), 2014.
[13] PyTorch – Open Source Machine Learning Library, Available: https://pytorch.org.
[14] Amazon Product Review Dataset, Stanford SNAP Project, Available: https://snap.stanford.edu/data/amazon.