Journal of Theoretical and Applied Information Technology
15th April 2024. Vol.102. No 7

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DESIGN OF RECOMMENDENDATION SYSTEMS USING DEEP REINFORCEMENT LEARNING – RECENT ADVANCEMENTS AND APPLICATIONS

KRISHNAMOORTHI.S[1], GOPAL K. SHYAM[2]

[1]Research scholar, Department of CSE, Presidency University, Bengaluru, India
[2]Professor, Department of CSE, Presidency University, Bengaluru, India
E-mail: [1]krishnamoorthis@gmail.com, [2]gopalshyambabu@gmail.com

ABSTRACT

The paradigm of recommendation systems (RS) has witnessed remarkable evolution in terms of providing accurate recommendations to the users. However, it is a complex task to generate appropriate recommendations to the users. In this context, RS use Artificial intelligence (AI) based techniques to recommend products based on the customer's preference. The adaptability of these techniques suffer from complexities systems such as data availability, changes in the user preferences, and unpredictable items. This motivates the researchers to emphasize performance enhancement of RS by overcoming these problems. This review focuses on the implementation of deep reinforcement learning (DRL) algorithms for RS. The study discusses different design aspects of RS and summarizes DRL-based techniques applied for recommendation systems. In addition, this review analyzes the challenges and relevant solutions based on the existing literary works. This paper also discusses the open issues of DRL and highlights the potential research directions in the RS field.

Keywords: *Comparative Analysis, Deep Reinforcement Learning, Policy Optimization Algorithms, Recommendation Systems*

1. INTRODUCTION

1.1 Background of Recommendation Systems

Recommendation systems (RS) are considered to be an essential part for most of the E-commerce and online systems. The evolution of social web applications has elevated the need for online service [1]. The RS models work considering the current interest of the user and do not emphasize on their long-term preferences. This is mainly due to the frequent variations observed in the user's interests which changes over time based on their interests, actions, preferences, and requirements. This dynamic behavior of users affects the functioning of recommendation systems. Due to this fact, most of the RS are designed considering short term interests of the users. Recently, online service platforms are using advanced and learning-based models for selling their products online by generating personalized recommendations to the users [2]. Ample research works have analyzed the necessity of developing an effective RS which can predict the interests of the users. However, it is not an easy task to generate recommendations tailored for customers.

In this context, online platforms are using Artificial intelligence (AI) to recommend or suggest products by interpreting customer preferences. Conventional machine learning algorithms (ML) fail to interpret large scale data without adequate training. On the other hand, deep learning-based RS are not effective in capturing interest dynamics since they are trained on the existing dataset which might not define real-time user preferences that change rapidly. The process of deep reinforcement learning (DRL) is different from machine learning and deep learning models in terms of its ability to learn through the agent by directly interacting with the external environment, without requiring any exemplary supervision. Since it learns directly from the environment, DRL-based models can make appropriate decisions and manage dynamic user preferences when implemented for generating recommendations. Hence the attention is shifted to DRL-based recommendation system (DRL-based RS) for generating recommendations and to improve long-term predictions. However, the dynamic variation and uncertainty in the user preferences and interests makes it complicated to suggest products. This motivates the researchers to focus more on improving the recommendation quality by

overcoming these problems. This research presents a detailed evaluation on the design and application of RS using DRL.

This review article focuses on discussing the emerging topics, open issues, challenges, and research gaps observed from existing literary works related to DRL-based RS and provides a clear perspective on this advancing domain. The novel contributions of this review article are as follows:

- This survey provides a comprehensive analysis on the design aspects and considerations of DRL-based RS including important RL algorithms for their policy optimization.
- This article provides the summarization and comparative analysis of DRL-based RS which includes the summary of evaluation metrics and provides detailed insights of selected journal publications.
- This survey emphasizes policy optimization algorithms such as TRPO, PPO, and DDPG for optimizing the RS.
- This survey presents an empirical analysis of the challenges and issues associated with DRL-based RS.

1.2 Research Significance

RS is investigated extensively in recent times. Extensive literature review has been conducted by various researchers in this aspect. The work presented in [3] reviewed the state of art of cross domain RS (CDRS) which can generate recommendations based on the user's interests. This work fulfills the gap demanding a systematic survey of CDRS using deep learning algorithms. However, the existing studies are restricted to the evaluation of recommendation systems using deep learning algorithms only. There is a collective demand for a literature review that extends its research scope beyond deep learning algorithms. An attempt to overcome this drawback is done by the work presented in [4] which provides a systematic review of DRL for RS. This review article discussed the motivation behind the application of DRL for recommendation systems. Existing works reviewed different DRL-based RS and summarized the existing techniques. However, the review focuses only on the analysis of DRL algorithms, evaluation and comparison of different DRL models such as single agent models, multi agent models, hybrid models etc and does not discuss the complexities associated with the design aspects.

1.3 Research Design Considerations

The review paper is designed considering the significance of the DRL in the design of RS. For structuring the review, several relevant articles were sourced based on the keywords and search strings. The articles were assessed from different search engines such as Elsevier, Springer, Research Gate, Journals and Conference papers related to the design of RS and Google scholar. In addition to this, the keywords are formulated using Google search trends. These sources are considered as the most valuable sources used for obtaining high quality research articles and journals. The relevant articles are sourced from the electronic database.

The articles were filtered using multiple filtration criterions. Based on the criteria, the articles were included or excluded from the review. In the first stage of filtration, the articles related to DRL-based RS were collected from the search engines and online databases using the keywords. All keywords were considered in this search. In the second stage of filtration, the articles were selected only if the papers had keywords and strings. In the third stage of filtration, the articles based are excluded on year of publication and journal of publication i.e., papers older than 2004 were not considered for the review. In the last and fourth stage of filtration, the articles were sourced based on the abstract i.e, if the abstract is relevant to the study, then the articles are considered, else the articles were excluded. After filtering, an overall 50 papers were finalized for conducting the review.

The criterions applied for selecting the articles to review helped in the precise interpretation of the results since they provide valuable insights about the design aspects of DRL, mechanism and application in the design of RS. In addition, the research design and selection criteria helped in providing a coherent narrative, making it easier to identify overarching patterns and draw accurate conclusions.

The review article is further organized as follows: Section 2 presents a brief overview of different types of DRL and their working process. Section 3 discusses the design aspects and considerations of DRL-based RS. Section 4 discusses the summarization and different comparative analysis of DRL algorithms, and Section 5 outlines the challenges and issues associated with DRL-based RS. Lastly, Section 6 concludes the paper with value-added information extracted from the existing literature works.

2. DEEP REINFORCEMENT LEARNING (DRL)

The DRL algorithm is one of the efficient learning-based algorithms which outperforms ML models in terms of computational performance, accuracy, and ability to process complex data patterns [5]. The DRL technique employs an efficient Q-learning mechanism which allows the system parameters to make decisions automatically without requiring any previous knowledge of the environment. Q-learning does not require any policy for learning and it learns from the actions of the model. The ability of Q-learning allows the system to take appropriate actions by observing the system environment. This reduces the requirement of additional resources for training the algorithm.

The RL algorithm uses an agent-environment interface for modeling the reinforcement problem as illustrated in figure 1.

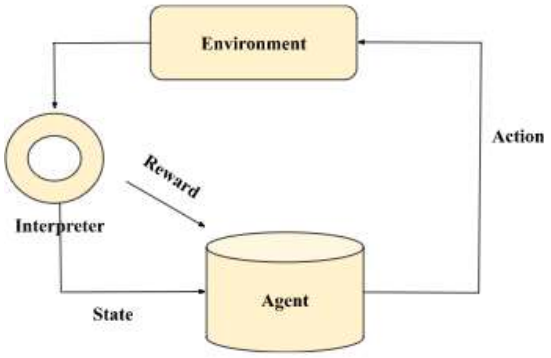

Figure 1: The agent-environment interface for modeling RL

In RL modeling, the agent-environment interface consists of an agent and its action. Here, the agent is termed as the learner of the decision maker and the environment is where the actions take place. For a time step 't', the agent senses or observes the state of the environment and based on the current state the model performs an action. The action is interpreted by the interpreter and for every action the model receives a reward from the environment. Typically, the RL problem is computed as a Markov Decision Process (MDP) and is defined as a tuple denoted by (S, A, R, P, γ), where S and A are defined as the set of all possible states and available actions for each state respectively, R signifies the reward, P is defined as the probability of transition, and γ represents the discount factor and cumulative reward as shown in equation 1:

$$\max E|R_{(T)}| \ldots\ldots (1)$$
$$\text{Where } R_{(T)} = \sum_{t=0}^{\tau} \gamma^t r(a_t, s_t)$$
$$0 \leq \gamma \leq 1$$

The primary components of the RL model are policy, reward shaping, value function, and model.

(i) Policy: The term policy in RL is denoted by π which evaluates the probability of taking an action 'a' for a given state 's'. Based on the performance, the RL policies are categorized as on-policy and off-policy methods. In on-policy methods, the RL model aims to improve the policy in order to make better decisions. On the other hand, off-policy methods use a policy which is different from the actual policy used for generating the information.

(ii) Reward Shaping: In RL tasks, the process of reward shaping is carried out to obtain appropriate localized advice for achieving maximum expected, discounted reward in a MDP process. Reward shaping is a process of embedding the knowledge of the environment into an RL model in order to train the algorithm to achieve faster and accurate solutions.

(iii) Value function: The value function validates the action taken by the RL model as good or bad action at a longer run.

(iv) Model: The model provides an accurate analysis about the behavior of the environment for a particular state.

2.1 Algorithms for solving RL problem

Several algorithms are proposed in various existing literary works for solving the RL problem. These algorithms are categorized as tabular and approximate processes. In the tabular process, the value functions are tabulated in the form of tables. Dynamic programming, Monte Carlo (MC), and temporal difference (TD) are some of the prominent tabular methods. The dynamic programming (DP) based models assume that the RL model is appropriate for the environment and hence use a value function for searching good policies. The DP models incorporate two prominent algorithms known as policy iteration and value iteration. The MC based methods do not assume that the environment is suitable for the RL model and hence they need complete information about S, A, and R from the environment. On the other hand, the TD based methods integrate the functionalities of both the DP and MC processes. However, the TD models can update the information by itself without requiring any information about the environment. One of the prominent algorithms for solving RL problems is the Q-learning and SARSA algorithms which follow the off-policy and on-policy process respectively.

Another category of algorithms is the approximate techniques which approximate the size of the state and the environment. The approximate techniques identifies an appropriate approximate solution without increasing the computational complexity and number of resources. Policy gradient methods is one of the excellent approximate solutions which uses a parameterized policy for learning and decides the actions without depending on the value function. And among policy gradient methods, REINFORCE and actor-critic [6] are the most prominent algorithms.

Deep learning (DL) algorithms based on neural network architecture have attracted the researchers because of its excellent performance. The sounding performance of the DL algorithms has motivated the researchers to incorporate the intelligence of DL algorithms into traditional RL algorithms and develop a deep Q-network (DQN), which is used as an approximate technique for Q-learning. Further this idea is extended in another policy gradient algorithm called as deep deterministic policy gradient (DDPG) which is a combined form of DQN and conventional DPG.

2.2 Comparative analysis of DRL applications

The application of different DRL algorithms are summarized in this section.

Table 1. Algorithms for problem solving in reinforcement learning models.

References	Category	Algorithm used	Applications
[7]	**Reinforcement Learning (RL)**	Temporal difference (TD) (Q-Learning and SARSA)	Estimation, Smooth Approximation, And Optimal Placement
[8]		Monte Carlo (MC)	Real-time online learning, Enhancement of the convergence speed and performance of DQN,
[9]		Dynamic programming (Value/policy iteration)	Solving decision making problems and convergence problems.
[10]	**Deep Reinforcement Learning (DRL)**	Q-Learning (DQN)	Achieving accurate forecasting, reducing computation complexity and improving state representation
[11]		Actor-Critic	To learn a safe and non-conservative policy, to mitigate model bias, and to reduce the physical interaction with external environment
[12]		REINFORCE	To achieve sub-optimal policy in two-stage recommendation systems, and analyzing user profile for personalized recommendations

Table 2. Summarization of different DRL algorithms

Reference	RL algorithm	State	Action	Reward	Evaluation method
[13]	Q- Learning	K-tuples	Approximation of Bayesian Interface	Prediction Accuracy	Offline method
[14]	SARSA	Current resource consumption state	Task Offloading and Resource Allocation	Smaller Response Time	On-line policy learning
[15]	SARSA (λ)	Series of discrete time steps	Interaction between the Agent and Environment	High learning efficiency	On-line policy learning
[16]	Value Iteration	Input /Output (IO) data	Closed loop control action	Optimal control	Optimal MDP policy
[17]	Policy Iteration	A set of variables obtained from interaction with environment	Controlling RL components	A possible reward for optimization problem	Offline training
[18]	Q- Learning & SARSA	Flappy Bird game	Performing flap and "do not flap" action	Positive reward for continuous action	Offline simulation and online

The emergence of DRL models has been a turning point in the design of recommendation systems. One of the outstanding abilities of DRL algorithms is to solve complex high dimensional state and action spaces which is most common in RS with large state and action spaces. As inferred from existing works, DQN is considered to be the most popular algorithm. DQN transforms the operation of traditional Q-learning algorithms by employing an experience relay, for updating weights in training phase, by reducing the computational complexity and by mitigating the effect of error derivatives. These factors help DQN achieve high stability compared to Q-learning. However, there are certain limitations which restrict the adaptability of DQN such as; overestimation of action values for certain situations makes it an inefficient learning algorithm and can result in suboptimal policies. To overcome this problem, DDQN algorithm is used in various recommendation systems [19]. Another limitation of DQN is that it randomly selects the experiences irrespective of their significance which affect the speed of the learning process. This problem is alleviated using policy gradient algorithms which not a value need function for approximation. The most popular policies are actor-critic and REINFORCE methods. These policies can update the policy weights directly. However, these methods suffer from high variance and slow learning problems.

3. DESIGN ASPECTS OF DRL-BASED RS

The design of DRL-based RS is discussed using three main stages: phases in RS, types of RS, and optimization of RL algorithms for RS.

3.1 Phases involved in RS

For recommending any item or a product, the RS uses three important stages which are discussed in below points:

(i) Information Collection Phase: In general, the recommendation system extracts a huge amount of information from large scale datasets. The information collected is related to user interests, preferences and ratings.

(ii) Learning Phase: The DRL algorithm is applied for the data collected from the previous phase and is processed by filtering the irrelevant and redundant information. Further, the user's features are applied for generating relevant recommendations.

(iii) Prediction or Recommendation Phase: In this stage, the DRL algorithm is trained to predict the items or products based on the observed activities of the user, preferences and interest.

The learning and prediction phases combined together helps the DRL algorithm to generate relevant recommendations. However, the learning process for each RL algorithm is different and based on this, the DRL-based RS are broadly categorized into model-based (MB), and model-free (MF) methods.

3.2 Types of DRL-based RS

3.2.1 Model-based Methods:

The MB approaches employ a learning model for adapting themselves to new tasks in complex environments and in real-time scenarios. In model-based methods, if the values of the dynamics of the models p ($x_{t+1} \mid x_t , u_t$) are known, then it is simple to approximate them using the learned model $\hat{Q}(x_{t+1} \mid x_t , u_t)$ which can be used for optimal control. For a linear dynamic function and quadratic rewards, the function Q (x_t , u_t) and V (x_t) defines the action-value function and value function respectively. These functions are computed using dynamic programming.

The model-based approaches have been broadly researched and there have been a list of publications which discussed the effectiveness of these approaches, which are tabulated in table 3.

Table 3. Model-based DRL for recommendation systems

Method	Reference
Value-based Method	[20]
Policy-based Method	[21]
Hybrid Method	[22]

- ***Challenges/Issues associated with Model-based DRL for RS***

It is inferred from existing studies that the performance of model-based DRL for RS deteriorates when implemented for handling large scale data. In value based methods, exploration is necessary to learn the stochastic values of the systems. These models with learned dynamics are affected by local minima which is more adverse than using ground-truth dynamics. Besides, the prediction or recommendation error increases with time for unknown conditions. Hybrid methods which integrate the attributes of both value-based and policy-based methods require more exploration in terms of its adaptability in recommendation systems.

3.2.2 Model-free Methods:

The MF approaches are more efficient in learning complicated environments, but it requires a greater number of iterations for achieving convergence which leads to local minima. These algorithms help to solve complex and high-dimensional problems. Also, model-free algorithms overcome the problem of requiring large numbers of samples as required by model based algorithms. However, these processes require a relatively large number of samples. In another case, off-policy algorithms employ Q-function approximation for obtaining superior data efficiency. The Q-function Q^{π} (x_t , u_t) for a defined policy π is determined for an expected return award from x_t after performing action u_t and following the generated policy π: The Q-function is defined as follows:

$$Q^{\pi}(x_t, u_t) = E_{rt} \geq t, x_t > t \sim E, u_t > t \sim \pi[R_t | x_t, u_t] \ldots\ldots (2)$$

The Q-learning learns a greedy deterministic policy $\mu(x_t) = argmax_u Q(x_t, u_t)$ which corresponds to the function $\pi(u_t|x_t) = \delta(u_t = \mu(x_t))$.

The review of model-free approaches for DRL-based RS are discussed in table 4

.

Table 4. Model-free DRL for recommendation systems

Method	Technique used	Reference
Value-based Method	Vanilla DQN	[23]
	Appropriate state and action optimization	[24]
	DQN with graph/image input	[25]
	DQN for joint learning	[26]
Policy-based Method	Vanilla REINFORCE	[27]
	REINFORCE uses graph structure/input	[28]
	Non-REINFORCE model	[29]
Hybrid Method	Vanilla DDPG	[30]
	DDPG with knowledge graph	[31]

The learning ability of policy-based methods is advantageous compared to value-based methods, since they do not need a value function for learning. In addition, policy based methods are simpler, and more deterministic than value based methods. On the other hand, hybrid methods combine the advantages of both action and value based methods are also gaining huge significance because of their ability to solve continuous action problems. It can be inferred that, DDPG with knowledge graph performs better compared to Vanilla DQN, DQN with graph input and joint learning. This is mainly because, DDPG is an actor-critic model that learns the policy directly from model parameters whereas DQN learns the Q values which define the policy. Training DDPG can be really challenging because of its unstable learning characteristics. In addition, DDPG is computationally intensive compared to DQN. This validates the effectiveness of DQN in deterministic tasks. However, it is not practically proved that DQN performs better than DDPG in discrete tasks. This aspect requires more exploration and validation.

Very few literary works on DRL-based RS have focused on policy based methods and hybrid methods such as DDPG. Most of the survey papers have focused on value function approaches, MDP, knowledge graphs etc. This work emphasizes policy optimization methods to address this research gap.

3.3 Recommendation Policy optimization using RL algorithms

In general, the DRL-based RS suffer from certain unique problems such as reward estimation, state construction, and simulation of external environments. To overcome these problems, DRL algorithms are optimized using on-policy and off-policy optimization algorithms. Three important policy optimization algorithms are discussed in this section. The DDPG algorithm can overcome the

drawbacks of DQN in terms of better control performance, while TRPO improves the performance of DDPG by ensuring a long-term reward. PPO further optimizes TRPO by transforming the surrogate objective

function, which enhances the efficiency and reduces the computational complexity.

3.3.1 Trust Region Policy Optimization (TRPO) Algorithm:

The TRPO is one of the prominent On-policy reinforcements learning based techniques which is adopted for updating the state of the system using the data generated by the present state of the system. The TRPO algorithm is mainly used for performing optimization of various large nonlinear components such as neural networks. This algorithm assures the theoretical monotonic improvement by estimating the gradient of the expected return $\bar{\nabla}_\Theta\, \eta\, (\pi_\Theta)$ using likelihood ratio as shown in below equation:

$$\overline{V_\theta}\eta(\pi_\theta) = \frac{1}{NT}\sum_{i=1}^{N}\sum_{t=0}^{1}\overline{V_\theta}\, log\pi_\theta(a_t^i|s_t^i)(R_t^i - b_t^i)\ldots. (3)$$

Where N is defined as the number of iterations, T is the total number of steps involved in every iteration, $R_t^i = \Sigma_{t'=t}^{T}$, $b_t^{t'-t}\, r_{t'}^{,i}$ is the cumulative reward function and b_t^t represents the variance for narrowing the baseline. The potentiality of the TRPO is maximized by employing reinforced learning which guarantees transformation in the policy distribution.

3.3.2 Proximal Policy Optimization (PPO) Algorithm:

The PPO technique acting as TRPO attempts to predict the ascent direction of the policy gradient of an expected return. This limits the changes in the operation of the policy to very small values. The policy gradient methods operate by evaluating the estimator of the policy gradient and by transforming it into a stochastic gradient ascent algorithm. The estimator used for representing the gradient is defined in the form as shown in below equation:

$$\hat{g} = \hat{E}_t[\overline{V_\theta}\, log\pi_\theta(a_t|s_t)\widehat{A_t}\ldots. (4)$$

Where π_Θ is defined as the stochastic policy and $\hat{A}_t$ represents the estimator which belongs to an advantage function for a time period t. The term $\hat{E}_t$ denotes the empirical average for a definite set of instances in PPO that switches between optimization and sampling. The implementation of PPO algorithm using an automatic differentiation mechanism is realized by optimizing the objective function of the estimator ĝ as shown in equation 5.

$$L^{PG}(\theta) = \hat{E}_t[log\pi_\theta(a_t|s_t)\widehat{A_t}\ldots. (5)$$

A PPO algorithm which employs the trajectory segments of fixed length is given in the algorithm shown below. During every iteration, each N parallel element aggregates the T time steps of data and the surrogate losses are computed based on the NT time steps of data which are further optimized using stochastic gradient descent algorithm (SGD).

PPO Algorithm:

Start:
for iteration = 1,2 …. **do**
for acto = 1,2, …. N **do**
Run policy $\pi\theta_{old}$ in environment for T time stamps
Compute advantage estimates $\widehat{A_1}, \widehat{A_2}$, …. $\widehat{A_T}$
end for
Optimize surrogate L wrt K epochs and
mini batch size M ≤ NT
$\theta_{old} \leftarrow \theta$
end for

3.3.3 Deep Deterministic Policy Gradient (DDPG) Algorithm:

Unlike, on-policy optimization algorithms, off-policy techniques allow learning based on all the available data from random and inconsistent policies. Off-policy techniques augment the efficiency of the learning based algorithms relative to on-policy based processes. One such efficient off-policy technique is the DDPG algorithm which is used for constructing a MF based RL model.

The DDPG algorithm is an actor-critic conventional reinforced learning approach. The problem of this algorithm is proved by the policy gradient theorem for a stochastic policy π (s, a ; θ) as shown in below equation:

$$\Delta\theta = \alpha\frac{\partial\rho}{\partial\theta} = \alpha\sum_s d^\pi(s)\sum_a \frac{\partial\pi(s,a)}{\partial\theta}\, Q^\pi(s,a)\ldots.. (6)$$

Where α is defined as the positive step size and d^π denotes the discounted weight of the states that starts at s_o. The actor-critic relationship of the DDPG is explained using below expressions: The actor π (s; θ^π) of the RL network layer depends on the present state of the environment s and consists of weights θ^π and the critic of another network of reinforced learning is denoted as Q (s, a; θ^Q). The critic of the network is updated using the Bellman equation as shown in below equation:

$$Q(s_t, a_t) = E_{rt}, S_{t+1}[r(s_t, a_t) + Q(S_{t+1}, \pi(S_{t+1})]\ldots.. (7)$$

The actor is updated using the chain rule and the weights θ^{π} are updated using the gradient loss function as shown below:

$$\widehat{V_{\theta\pi}}L \approx E_s[\widehat{V_{\theta\pi}}Q(s, \pi(s|\theta^{\pi})|\theta^{Q}\ldots. (8)$$

$$E_s\left[\widehat{V_a}Q(s, a|\theta^{Q})\middle|a = \pi(s|\theta_{\pi})\overline{V_{\theta\pi}}\pi(s|\theta^{\pi})\right] (9)$$

The algorithm for the DDPG mechanism is given below.

Algorithm: DDPG algorithm:

Start:
Initialization
Input weights of Q $(s, a \mid \theta^{Q})$ and $\pi\ (s \mid \theta^{\pi})$
Output weights $\theta^{Q'} \leftarrow \theta^{Q}, \theta^{\pi'} \leftarrow \theta^{\pi}$
replay buffer R
for each process **do**
Initialize process N to explore state actions
Observe initial state S_1
for each step t of process **do**
Select action $a_t = \pi(s_t|\theta^{\pi}) + N_t$
Compute a_t and obtain reward r_t and state S_{t+1}
Obtain state transition (S_t, a_t, r_t, s_{t+1})in R
Evaluate the samples stored in R
Set $y_i = r_i + \theta Q'(S_{i+1}, \pi'(S_{i+1}|\theta^{\pi'})|\theta^{Q'}$
Update critic by optimizing the surrogate loss
$L = \frac{1}{N}\sum_i(y_i - Q(s, a|\theta^{Q}))^2$
Update the actor using the sampled policy gradient
$\overline{V\theta_{\pi}}L \approx \left(\frac{1}{N}\right)\sum_i \overline{V}_a Q(s, a|\theta^{Q})\overline{V\theta_{\pi}}\ \pi(s|\theta^{\pi})|s_i$
Update the systems:
$\theta^{Q'} \leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'}$
$\theta^{\pi'} \leftarrow \tau\theta^{\pi} + (1-\tau)\theta^{\pi'}$
end for
end for

Several researchers have discussed the prominence of TRPO, PPO, and DDPG algorithms for application of the above discussed optimization algorithms. Table 5 summarizes the application of TRPO, PPO, and DDPG algorithms for different RS.

It can be inferred from the literature review that PPO and TRPO algorithms both achieve better performance compared to other gradient based methods. However, PPO outperforms TRPO in terms of fine tuning the DRL parameters, consistency in solving policy iterations and high CTR. In addition, both these algorithms can be trained faster with lesser parameters for generating recommendations. On the other hand, DDPG achieves better performance and stability compared to the DQN networks. Due to learning ability, DDPG algorithms are more suitable for generating automatic recommendations with better NDCG compared to other deep learning based models. The literature review also reveals that the TRPO, PPO, and DDPG algorithms yield better results compared to value based methods and neural network models. This necessitates the need for deeper investigation of these algorithms for RS.

Table 5. Policy optimization algorithms for different RS

Reference	Algorithm used	Application	Evaluation Metrics and Values	Observation
[32]	PPO	E-Commerce	CTR = 2.843%	Improvement in CTR rate compared to model-based and model-free methods
[33]	TRPO	Online learning systems	AUC = 0.739	Prediction capacity increased 3 times compared to LSTM and context-aware models
[34]	DDPG	Page-wise recommendations	F1-score = 80.5%, and NDCG = 0.1872	Automatic learning of recommendation strategy compared to deep CNN, and GRU.
[35]	DDPG	Interactive RS	NDCG = 0.8 and Precision = 0.45	Improvement of 40% and 30 % in terms of precision and NDCG is observed.

[36]	TRPO	E-Commerce	CTR = 0.33 (at 30k step)	TRPO requires less training time than DQN and TRPO achieves better performance in terms of CTR and single item recommendations
[37]	PPO	Online simulated environment named Virtual Taobao	CTR = 0.37	PPO achieves better CTR with stable performance compared to DDPG

4 SUMMARIZATIONS OF DRL-BASED TECHNIQUES APPLIED FOR RECOMMENDATION SYSTEMS

This section provides the summarization and different comparative analysis of DRL-based RS. A comparative table and summarization is tabulated to discuss different DRL algorithms proposed by different researchers for RS.

Table 6. Overview of DRL-based recommendation approaches

Ref	Recommendation Method	Advantages	Limitations
[38]	Collaborative filtering	High recommendation accuracy and faster convergence	Suffers from state space problems and requires high computation time
[39]	Collaborative filtering	Provides better results for large scale user database, No prior knowledge is required, Improved recommendation time	Data sparsity problem and cold start problems while recommending new items
[40]	Learning based Recommendation	Performs feature interaction modeling for identifying user interaction.	The model capacity becomes limited after reaching peak performance
[41]	Long-term Recommendation	Maximizes recommendation accuracy with high hit rate and NDCG	The performance of the RL algorithm is affected due to sampling efficiency problems
[42]	Cold-start recommendation	Overcome the cold-start problem in RS and achieves high CTR and NDCG	The proposed approach focuses only on cold-start problems and is not designed for lifelong recommendation
[43]	Learning based Recommendation	The model observes improved convergence rate and is able to compute complicated and high-dimensional instances	The model is generic and underperforms while generating application specific recommendations
[44]	Cross-platform recommendation systems	Overcome the problem of cold-start, gray sheep, and data sparsity problems	Reduction in scalability and increase in the computation cost is observed with the increase in the diversity of social networks and number of users.
[45]	Shared-account Cross-domain Sequential Recommendation	The SCSR is advantageous compared to cross platform RS since it considers the characteristics of both the shared-account and	The SCSR model assumes that all accounts are shared by the same users, which is not appropriate for real-time applications. In real-time shared

	(SCSR)	cross-domain while generating recommendations for shared-accounts.	accounts, both the identity and number of the latent users are not known.

4.1 Performance Evaluation of DRL-based RS

The performance of DRL-based RS is evaluated using both offline and online evaluation methods [45]. During offline evaluation, the efficacy of the DRL-based RS is evaluated for a fixed dataset using metrics such as RMSE, MAE, precision, recall, F-measure, AUC, NDCG, MAP, hit rate, and perplexity Whereas in the online evaluation process, the DRL-based RS is tested for its ability to learn while interacting with the environment. CTR and bounce rate are the two main parameters which are used to determine the online performance of DRL-based RS. The different performance metrics used for DRL-based RS are discussed in table 7.

Table 7. Summary of evaluation metrics for evaluating DRL-based RS

Reference	Evaluation Metric	Description	Mathematical Formula
Offline Evaluation			
[45] [46]	Root Mean Squared Error (RMSE)	RMSE is used in predictive and regression analysis. It is equal to the square root of the MSE.	$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n} e_t^2}$
[47]	Mean Absolute Error (MAE)	MAE is calculated as the average of absolute difference between the predicted and actual values	$(\text{MAE}) = \frac{1}{n} * \sum_{i=1}^{N} \lvert x_i - x \rvert$
[46]	Precision	Precision is defined as the accuracy of positive predictions. It is also defined as the proportion of accurately identified data which is relevant.	$\boldsymbol{Precision = \frac{TP}{TP + FP}}$ Where TP and FP are true positive and false positive respectively.
[47]	Recall	Recall is determined as the ratio of the recommendations that are accurately classified	$\boldsymbol{Recall = \frac{TP}{TP + FN}}$ Where TP and FN are true positive and false negative respectively.
[47]	F-measure	F-measure is determined as the ratio of mean of its precision and recall.	$\boldsymbol{F\ measure = \frac{2 * Precision * Recall}{Precision + Recall}}$
[46]	Area Under the Curve (AUC)	AUC is the ability of the RL model to distinguish different samples in a particular dataset	NA

[47]	Normalized Discounted Cumulative Gain (NDCG)	The NDCG measure is used for ranking the products based on their accuracy.	$nDCG_p = \frac{DCG_p}{IDCG_p}$
[47]	Mean Average Precision (MAP)	MAP is computed as the mean of the average precision value based on the relevance score.	$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i$ Where AP is the average precision
[48]	Hit rate	The total number of products in the samples that are also present in the user list is defined as the number of hits. And the number of hits defines the hit rate.	NA
[56]	Perplexity	This metric is used for evaluating the topical models which can measure the quality of the recommended items.	$PP(W) = \sqrt[N]{\frac{1}{P(w_1, w_2, \ldots . w_N)}}$ Where w_1, w_2 ….w_N represent the words
		Online Evaluation	
[49]	Click Through Rate (CTR)	CTR is the measure of the recommendations that are clicked by the user	$\text{CTR} = \frac{\text{Total no of clicks}}{\text{Total no of impressions}} * 100$
[50]	Bounce Rate	Bounce rate is defined as the total percentage of users who verify the recommendation list and do not explore the recommendations and exit the system.	$\text{Bounce rate} = \frac{\text{Single page visits}}{\text{Total website visits}}$

Hit-Rate and NDCG are the two main evaluation metrics that are used to evaluate the performance of DRL-based RS used for generating long term recommendations. For real-time evaluation, the CTR can help since it provides the accurate measurement of the recommendations.

5. CHALLENGES AND ISSUES IN DRL-BASED RS

Despite the advantages there are certain challenges and issues which need to be addressed. Existing DRL-based RSs have been found to face accuracy-related issues, cold-start issues and sparsity issues when dealing with large scale data. Apart from traditional issues such as cold-start and data sparsity problems, there are certain challenges in these systems which need significant attention. Some of the prominent research problems identified in this work are listed in below points:

(i) Sample Efficiency: Sample inefficiency is one of the crucial issues in MF-based DRL methods. The number of samples required to train MF-DRL is significantly high and the agent interacting with the external environment requires sufficient training data to perform actions. However, the sample efficiency of the model-based (MB) DRL is comparatively higher compared to MF-DRL. However, MB-DRL systems are more complex since the agent is forced to learn both the policy and the external environment.
(ii) Data Dimensionality: Since recommendation systems have to deal with complex and large-scale data, they often suffer from the problem of class imbalance and data dimensionality which

deteriorates the recommendation accuracy of the DRL models.

(iii) Optimization of DRL: Optimizing the goals of DRL for multi-objective RS involves a lot of complexities in terms of ensuring fair recommendation generation, increased computational cost, and maintaining diversity.

(iv) Biased and Sparse feedback data: In most of the RS, the feedback is usually biased. In DRL-based RS the

feedback is sampled based on the interaction between the policy and the environment. Though off-policy training can enhance the policy based on the biased feedback data, the sample efficiency is reduced significantly. Besides, the RS is designed to handle millions of users with indistinct choices which makes the DRL user state space more complex to interpret.

(v) Online deployment: Deploying DRL-based RS for handling multiple scenarios and multiple customers at a time can be challenging. Training DRL models while achieving high relevance and CTR is crucial and requires robust training. This issue can be resolved by deploying an online training model. However, online deployment is still in the infant stage and requires deeper investigation.

5.1 Assumptions and Limitations

Although this review paper attempts to cover all possible design aspects with challenges and issues related to DRL-based RS, there are certain assumptions and limitations that are undertaken in this study.

5.1.1 Assumptions

- This review assumes and generalizes the findings obtained across different recommendation systems. However, the interpretation of findings can vary based on the domain and application area.
- It is assumed while formulating this review that the algorithms and models discussed remain stable over time. Any small changes or updates in the model parameter can have a significant influence on the performance of the RS and accuracy.

5.1.2 Limitations

- The study mainly emphasizes the design of RS using DRL and performance evaluation. However, limiting the scope of this review may lead to potential research gap and necessitates the need for more relevant work.
- Considering the rapid evolution of technologies related to the design of RS, this review might become less informative and require frequent updates.
- One of the most important factors related to RS is the security, which is not emphasized in this review.

6. CONCLUSION

This review comprehensively analyzes the concept of RS with an emphasis on deep reinforcement learning (DRL) algorithms. The study discussed different DRL methods and were compared with respect to their state, action, reward, dataset used, evaluation method and metrics used for evaluation. The study also discussed the design aspects of DRL-based RS and their policy optimization algorithms. The analysis of TRPO, PPO, and DDPG algorithms shows that both TRPO and PPO algorithms achieve better recommendation performance in terms of prediction capacity, high CTR, and better learning ability. The DDPG algorithm exhibits better stability and recommendation performance compared to value-based methods. These policy optimization algorithms can be trained faster with limited parameters and hence are considered to be more appropriate for optimizing RS. More focus is required on the analysis of policy optimization methods for optimizing the performance of RS. It can be inferred from the review that the DRL algorithm has a huge significance in the field of RS and the optimization of RS needs a deeper investigation. The study also points out that the issue of data dimensionality can have a negative impact on the accuracy of the DRL-based RS. This issue can be resolved by integrating a feature extraction approach with DRL algorithms. Another prominent aspect of the RS is its susceptibility towards data bias and data dimensionality problems. These factors can pose a grave threat to the RS and hence focusing on solving them remains as a foremost challenge. It is expected that this review can provide a roadmap for the researchers to understand DRL algorithm and optimization algorithms and their problems and hence can provide valuable insight for the researchers carrying out research in this field.

REFERENCES:

[1] Vartak, M., Huang, S., Siddiqui, T., Madden, S., & Parameswaran, A. (2017). "Towards Visualization Recommendation Systems". *ACM Sigmod Record*, *45*(4), 34-39.

[2] Wang, K., Zhang, T., Xue, T., Lu, Y., & Na, S. G. (2020). "E-commerce personalized

recommendation analysis by deeply-learned clustering". *Journal of Visual Communication and Image Representation*, *71*, 102735.

[3] Gahier, A. K., & Gujral, S. K. (2021). "Cross Domain Recommendation Systems using Deep Learning: A Systematic Literature Review". *Available at SSRN 3884919*.

[4] Chen, X., Yao, L., McAuley, J., Zhou, G., & Wang, X. (2021). "A survey of deep reinforcement learning in recommender systems: A systematic review and future directions". *arXiv preprint arXiv:2109.03540*.

[5] François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., & Pineau, J. (2018). "An introduction to deep reinforcement learning". *arXiv preprint arXiv:1811.12560*.

[6] Millán-Arias, C. C., Fernandes, B. J., Cruz, F., Dazeley, R., & Fernandes, S. (2021). "A robust approach for continuous interactive actor-critic algorithms". *IEEE Access*, *9*, 104242-104260.

[7] Sewak, M. (2019). "Temporal difference learning, SARSA, and Q-learning". In *Deep Reinforcement Learning* (pp. 51-63). Springer, Singapore.

[8] Kim, C. (2020). "Deep reinforcement learning by balancing offline Monte Carlo and online temporal difference use based on environment experiences". *Symmetry*, *12*(10), 1685.

[9] Hamadouche, M., Dezan, C., Espes, D., & Branco, K. (2021, June). "Comparison of Value Iteration, Policy Iteration and Q-Learning for solving Decision-Making problems." In *2021 International Conference on Unmanned Aircraft Systems (ICUAS)* (pp. 101-110). IEEE.

[10] Carta, S., Ferreira, A., Podda, A. S., Recupero, D. R., & Sanna, A. (2021). "Multi-DQN: An ensemble of Deep Q-learning agents for stock market forecasting". *Expert systems with applications*, *164*, 113820.

[11] Morgan, A. S., Nandha, D., Chalvatzaki, G., D'Eramo, C., Dollar, A. M., & Peters, J. (2021, May). "Model predictive actor-critic: accelerating robot skill acquisition with deep reinforcement learning". In *2021 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 6672-6678). IEEE.

[12] Ma, J., Zhao, Z., Yi, X., Yang, J., Chen, M., Tang, J., ... & Chi, E. H. (2020, April). "Off-policy learning in two-stage recommender systems". In *Proceedings of The Web Conference 2020* (pp. 463-473).

[13] Cini, A., D'Eramo, C., Peters, J., & Alippi, C. (2020). "Deep reinforcement learning with weighted Q-Learning". *arXiv preprint arXiv:2003.09280*.

[14] Alfakih, T., Hassan, M. M., Gumaei, A., Savaglio, C., & Fortino, G. (2020). "Task offloading and resource allocation for mobile edge computing by deep reinforcement learning based on SARSA". *IEEE Access*, *8*, 54074-54084.

[15] Jiang, H., Gui, R., Chen, Z., Wu, L., Dang, J., & Zhou, J. (2019). "An Improved Sarsa ($\lambda $) Reinforcement Learning Algorithm for Wireless Communication Systems". *IEEE Access*, *7*, 115418-115427.

[16] Radac, M. B., & Borlea, A. I. (2021). "Virtual state feedback reference tuning and value iteration reinforcement learning for unknown observable systems control". *Energies*, *14*(4), 1006.

[17] Bertsekas, D. (2021). "Multiagent reinforcement learning: Rollout and policy iteration". *IEEE/CAA Journal of Automatica Sinica*, *8*(2), 249-272.

[18] Vu, T., & Tran, L. (2020). "FlapAI bird: training an agent to play flappy bird using reinforcement learning techniques". *arXiv preprint arXiv:2003.09579*.

[19] Liu, D., & Yang, C. (2019). "A deep reinforcement learning approach to proactive content pushing and recommendation for mobile users". *IEEE Access*, *7*, 83120-83136.

[20] Wang, K., Zou, Z., Deng, Q., Wu, R., Tao, J., Fan, C., ... & Cui, P. (2021, April). "Reinforcement Learning with a Disentangled Universal Value Function for Item Recommendation". In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 35, pp. 4427-4435).

[21] Hong, D., Li, Y., & Dong, Q. (2020, July). "Nonintrusive-Sensing and Reinforcement-Learning Based Adaptive Personalized Music Recommendation". In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 1721-1724).

[22] Zhao, X., Xia, L., Zou, L., Liu, H., Yin, D., & Tang, J. (2020, October). "Whole-chain recommendations". In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (pp. 1883-1891).

[23] Lei, Y., Wang, Z., Li, W., & Pei, H. (2019, July). "Social attentive deep q-network for recommendation". In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 1189-1192).

[24] Xiao, Y., Xiao, L., Lu, X., Zhang, H., Yu, S., & Poor, H. V. (2020). "Deep-reinforcement-

learning-based user profile perturbation for privacy-aware recommendation". *IEEE Internet of Things Journal*, *8*(6), 4560-4568.

[25] Zhou, S., Dai, X., Chen, H., Zhang, W., Ren, K., Tang, R., ... & Yu, Y. (2020, July). "Interactive recommender system via knowledge graph-enhanced reinforcement learning". In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 179-188).

[26] Zhao, X., Gu, C., Zhang, H., Yang, X., Liu, X., Liu, H., & Tang, J. (2021, May). "Dear: Deep reinforcement learning for online advertising impression in recommender systems". In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 35, No. 1, pp. 750-758).

[27] Chen, M., Chang, B., Xu, C., & Chi, E. H. (2021, March). "User response models to improve a reinforce recommender system". In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining* (pp. 121-129).

[28] Chen, H., Zhu, C., Tang, R., Zhang, W., He, X., & Yu, Y. (2021). "Large-scale Interactive Recommendation with Tree-structured Reinforcement Learning". *IEEE Transactions on Knowledge and Data Engineering*.

[29] Zhang, R., Yu, T., Shen, Y., Jin, H., & Chen, C. (2019). "Text-based interactive recommendation via constraint-augmented reinforcement learning". *Advances in neural information processing systems*, *32*.

[30] Liu, F., Tang, R., Guo, H., Li, X., Ye, Y., & He, X. (2020). "Top-aware reinforcement learning based recommendation". *Neurocomputing*, *417*, 255-269.

[31] Zhang, W., Liu, H., Wang, F., Xu, T., Xin, H., Dou, D., & Xiong, H. (2021, April). "Intelligent electric vehicle charging recommendation based on multi-agent reinforcement learning". In *Proceedings of the Web Conference 2021* (pp. 1856-1867).

[32] Zhang, J., Yin, J., Lee, D., & Zhu, L. (2019). "Deep reinforcement learning for personalized search story recommendation". Journal of Environmental Sciences (China) English Ed.

[33] Ai, F., Chen, Y., Guo, Y., Zhao, Y., Wang, Z., Fu, G., & Wang, G. (2019). "Concept-Aware Deep Knowledge Tracing and Exercise Recommendation in an Online Learning System". *International Educational Data Mining Society*.

[34] Zhao, X., Xia, L., Zhang, L., Ding, Z., Yin, D., & Tang, J. (2018, September). "Deep reinforcement learning for page-wise recommendations". In *Proceedings of the 12th ACM Conference on Recommender Systems* (pp. 95-103).

[35] Baghi, V., Motehayeri, S. M. S., Moeini, A., & Abedian, R. (2021, March). "Improving ranking function and diversification in interactive recommendation systems based on deep reinforcement learning". In *2021 26th International Computer Conference, Computer Society of Iran (CSICC)* (pp. 1-7). IEEE.

[36] Yu, Y., Gu, Z., Tao, R., Ge, J., & Chang, K. (2020). "Interactive Search Based on Deep Reinforcement Learning". *arXiv preprint arXiv:2012.06052*.

[37] Fang, W., & Zeng, F. (2020). "Policy Gradient for items Recommendation on Virtual Taobao".

[38] Hu, B., Shi, C., & Liu, J. (2017, October). "Playlist recommendation based on reinforcement learning". In *International Conference on Intelligence Science* (pp. 172-182). Springer, Cham.

[39] Bobadilla, J., Alonso, S., & Hernando, A. (2020). "Deep learning architecture for collaborative filtering recommender systems". *Applied Sciences*, *10*(7), 2441.

[40] Liu, F., Tang, R., Li, X., Zhang, W., Ye, Y., Chen, H., ... & He, X. (2020). "State representation modeling for deep reinforcement learning based recommendation". *Knowledge-Based Systems*, *205*, 106170.

[41] Huang, L., Fu, M., Li, F., Qu, H., Liu, Y., & Chen, W. (2021). "A deep reinforcement learning based long-term recommender system". *Knowledge-Based Systems*, *213*, 106706.

[42] Ji, L., Qin, Q., Han, B., & Yang, H. (2021, October). "Reinforcement Learning to Optimize Lifetime Value in Cold-Start Recommendation". In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (pp. 782-791).

[43] Zhao, D., Zhang, L., Zhang, B., Zheng, L., Bao, Y., & Yan, W. (2020, July). "Mahrl: Multi-goals abstraction based deep hierarchical reinforcement learning for recommendations". In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 871-880).

[44] Ke, G., Du, H. L., & Chen, Y. C. (2021). "Cross-platform dynamic goods recommendation system based on reinforcement learning and

social networks". *Applied Soft Computing*, *104*, 107213.

[45] Haruna, K., Akmar Ismail, M., Suhendroyono, S., Damiasih, D., Pierewan, A. C., Chiroma, H., & Herawan, T. (2017). "Context-aware recommender system: A review of recent developmental process and future research direction". *Applied Sciences*, *7*(12), 1211.

[46] Shani, G., & Gunawardana, A. (2011). "Evaluating recommendation systems". In *Recommender systems handbook* (pp. 257-297). Springer, Boston, MA.

[47] Kulkarni, S., & Rodd, S. F. (2020). "Context Aware Recommendation Systems: A review of the state of the art techniques". *Computer Science Review*, *37*, 100255.

[48] Deshpande, M., & Karypis, G. (2004). "Item-based top-n recommendation algorithms". *ACM Transactions on Information Systems (TOIS)*, *22*(1), 143-177.

[49] Dehghani Champiri, Z., Asemi, A., & Siti Salwah Binti, S. (2019). "Meta-analysis of evaluation methods and metrics used in context-aware scholarly recommender systems". *Knowledge and Information Systems*, *61*(2), 1147-1178.

[50] Helmers, C., Krishnan, P., & Patnam, M. (2019). "Attention and saliency on the internet: Evidence from an online recommendation system". *Journal of Economic Behavior & Organization*, *161*, 2