IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 27, 2022, accepted May 12, 2022, date of publication June 8, 2022, date of current version June 23, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3181164

Multiobjective Deep Reinforcement Learning for Recommendation Systems

EE YEO KEAT, NURFADHLINA MOHD SHAREF, RAZALI YAAKOB, (Member, IEEE), KHAIRUL AZHAR KASMIRAN, ERZAM MARLISAH, NORWATI MUSTAPHA, AND MASLINA ZOLKEPLI

Department of Computer Science, Universiti Putra Malaysia, Selangor 43400, Malaysia

Corresponding author: Nurfadhlina Mohd Sharef (nurfadhlina@upm.edu.my)

This work was supported by the Air Force Office of Scientific Research (AFOSR), United States Research Grant under Grant FA2386-18-1-4079.

ABSTRACT Most existing recommendation systems (RSs) are primarily concerned about the accuracy of rating prediction and only recommending popular items. However, other non-accuracy metrics such as novelty and diversity should not be overlooked. Existing multi-objective (MO) RSs employed collaborative filtering and combined with evolutionary algorithms to handle bi-objective optimization. Besides cold-start problem from collaborative filtering, it also vulnerable to highly sparse environment, while the evolutionary algorithm suffers from premature convergence and curse of dimensionality. These limitations have prompted this work to propose deep reinforcement learning (DRL) approaches for MO optimization in RSs. Several works in DRL are available but none has addressed MO RS problems. In this study, the performances of proposed DRL approaches that based on Deep Q-Network in MO recommendation problem were investigated. The approaches were evaluated with movie recommendation dataset by using three conflicting metrics, namely precision, novelty, and diversity. The results demonstrated that deep reinforcement learning approaches has superiority performance in MO optimization, and its capability of recommending precise item along with achieving high novelty and diversity against the benchmark that using probabilistic based multi-objective approach based on evolutionary algorithm (PMOEA). Although PMOEA algorithm secured higher average value in precision, it has lower values of novelty and diversity than the proposed DRL approaches. The DRL approaches surpassed the benchmark results in average of maximum novelty and the average of mean diversity metrics, the optimization between accuracy and non-accuracy metrics is inevitable. In addition, the experiments revealed that incorporation of user latent features enhanced the recommendation quality.

INDEX TERMS Deep reinforcement learning, machine learning, multiobjective, recommendation system.

I. INTRODUCTION

The volume of information and data are growing exponentially nowadays, where we can simply get tons of information through online applications at fingertips. Excessive information may cause the online users difficult to meet the user's interest or correct target information. Therefore, recommendation systems (RSs) are applied to direct users through vast information space, toward the items that could fulfill the user's desire. The recommendation algorithm is aimed to provide a list of relevant items that user might be interested in order to secure user satisfaction and maintain user activeness in the system. From business perspective, an RS is a crucial and valuable tool to boost the business revenue. For instance, an online streaming platform consists of few million movies and series and thus requires a recommendation engine to generate playlists anticipating their interest. When the suggested relevant items are matched with user interest, it is expected to increase the subscription rate and enhance the user's online streaming experience.

Approaches for RSs may be categorized into 6 classes [1] as follows:

i. Context-Aware RSs which provides references by user's contexts.

The associate editor coordinating the review of this manuscript and approving it for publication was Wentao Fan.

ii. Group RSs which consider the preferences of a group of users instead of a single user to generate recommendations.
iii. Multi-Criteria RSs which address user preferences various item's aspects (such as cleanliness and location, for the recommendation of a hotel).
iv. Cross-Domain RSs which utilizes knowledge inferred in a source domain (e.g., movie) for recommendation to users in a target domain (e.g., music).
v. Multi-Stakeholder RSs considers several perspectives of users to generate the recommendation.
vi. Multi-Task RSs which utilize ensemble method to combine output from several RSs typically through a joint optimization over a shared network representation.

Most of the traditional RS approaches [2]–[9] are only focused on accuracy of rating prediction or item with high rating. Recently, findings of several studies [10]–[13] have shown that non-accuracy metrics included novelty and diversity in RS are significantly correlated with the satisfaction level of users. Another interesting finding [14] shows diversification is one of the noteworthy factors that affect users' satisfaction positively. Other scholars [13], [15] also supported that favourable recommendation quality is highly correlated to novelty and diversity of the items. In other words, focusing solely on accuracy metric does not secure high-quality recommendation since the items with high accuracy result do not assure satisfaction of users [16].It is not argued that accuracy metrics should be neglected, but rather that other evaluation metrics should also be considered simultaneously, hence multi-objective (MO) based recommendation. It provides extensive motivation to this work in solving MO recommendation problem.

Among classical RS, content-based filtering (CB) [5], [6], [8] and collaborative filtering (CF) [2]–[4] are commonly used. According to [9], [17], the former approach endured the limitations of handling inter-dependencies event, whereas the latter struggled in cold-start problem and data sparsity [9], [17]–[23] to generate recommendation if it lacks sufficient historical relationship information between the user and item. Another review study [24] on social media analysis by machine learning indicated that typical RS algorithms such as matrix factorization or Support Vector Machine (SVM) also suffered from cold-start, serendipity, and scalability problems. In addition, the CB method encounters difficulty in suggesting items from categories that are new to the users or have not been experienced by them since it focuses only toward similar content or item group to user [25]. Furthermore, the techniques involving CF have the limitation to include side features for query item such as user's latent information. The hybrid approach [26]–[28] tackled some limitations of CB and CF approach, but the main weakness of dealing with a new user or item that has never been experienced still persists [29]. Moreover, traditional recommendation approaches fail to consider feedbacks from users [30] and are inadequate for handling MO problem.

Scalarization and population-based heuristics methods are the common techniques for MORS [1]. Scalarization is used to transform a MOP to a single-objective problem (SOP), so that most of the existing optimization methods for SOP can be reused to solve the problem. Meanwhile, the population-based heuristics utilizes evolutionary algorithms (EAs) or swarm intelligence methods (SIs) to produce the Pareto optimal set. Scalarization methods are popular due to its simplicity since they transform MOPs to SOPs. However, scalarization methods may not be able to handle non-convex problems in contrast to the Multi-Objective Evolutionary Algorithms (MOEAs). By contrast, MOEAs can handle both convex and concave problems but may suffer premature convergence at local optima besides the weakness in diversity and typically trapped in efficiency issue when processing large dataset.

Most of the existing MO optimization frameworks [31]–[33] integrate genetic algorithm (GA) with classical CF method despite them being time- and resource-consuming [34]. This poor reputation is due to abundance of iterations required to search and populate the solutions as it has constraints on scaling up for large-scale optimization task. GA is a prominent technique used among evolutionary computing (EC) or evolutionary algorithm (EA) approaches [35], and premature convergence is one of the critical weakness of EC approaches [36]. In GA, only the most optimal solution is selected and this hill-climbing-based solution leads to premature convergence problem. In addition, while using EC approach [37], it is difficult to achieve good density points and converge to optimal solutions [38].

There are also approaches that focus on personalization in MORS such as the preference-based method called the Extreme Dominance and Statistical Significance Tests for defining a new Pareto-based dominance relation that guides the optimization search considering users' preferences [39]. Multi-criteria RSs based on deep learning such as the deep autoencoders are employed to exploit the non-trivial, nonlinear and hidden relations between users with regard to multi-criteria preferences [40]. Tensor model has also been used in Multi-criteria RS [41] which combines aspects (e.g., users or countries, restaurants, multiple ratings, and cultural groups) and applies factorization (e.g. higher order singular value decomposition) to process the inter-relations of the various aspects for predicting the missing values in the models, and then used for predicting the rating.

In the last decade, reinforcement learning (RL) approaches have grabbed attention of many researchers as its applications are increasing progressively. RL approaches have rose to prominence for solving complex decision-making problems. Q-learning [42] is one of the basic RL techniques and has been widely researched in various fields including electric power management [43], the Internet of things [44], and RSs [45]. Despite its demonstrated ability to learn optimal strategy dynamically, tabular mapping function approach is inefficient in a high-dimensionality environment [46]. Therefore, various deep reinforcement learning (DRL) approaches such as Deep Q-Network (DQN) [47] and Deep Recurrent Q-Learning [48] have been proposed to overcome the shortcomings of basic RL tabular function approaches in large-scale environments.

Several researchers have demonstrated that DRL approaches such as Q-learning-based approach outperform the EC approach in various MO problems [49]–[51]. RL approaches in MO problems could optimize power consuming and voltage stability, and DRL approach is better than the EA in terms of Pareto solutions achievement; besides, DRL has more accurate optimal points [50]. A significant study [49] on multi-objective traveling salesman problem demonstrated that DRL approach outperformed the MO-based EC approaches in terms of solution convergence and large sparse data handling. They presented the proposed non-iterative solver and demonstrated that DRL approach is more efficacious than EC approaches for solving MO problem.

The characteristic of DRL to explore the environment and make decision autonomously has become very useful in RS applications. Several researchers have evidenced the practicality of DRL in complex RS environment with single-objective algorithms [52]–[55]. However, after a thorough search of relevant literature, it can be asserted that there is a lack of adequate research concerning the application of DRL techniques in MO problem of RS domain. In accordance with the works discussed above, the extensive potential of DRL approaches to handle MO problems in RS is further investigated throughout this work. This research further examines the capability of RL in RS application, along with MO optimization problem. In this context, we developed a DQN-based approach to solve MO problem in RS (called DQNMORS) and evaluated the same according to three metrics namely accuracy, novelty, and diversity. We have developed an MODRL approaches, which salient features are enumerated as follows:

1) Our approaches do not rely on rating predictor for MORS and optimize three evaluation metrics simultaneously, namely precision, novelty, and diversity. This paper also presents an extensive analysis of the comparison between optimization techniques by scalarization method and Pareto filtering method.
2) Our approaches are based on DQN (called DQNMORS), which incorporates user latent features to help improve recommendation and extensive investigation and show the impact of learning user latent on performance.
3) Our approaches consider time-sequential rating data as one of the input types and study the impact of learning sequential rating data using recurrent layer (through a model called recDQNMORS).

The remainder of this paper is organized as follows. Section 2 discusses literature review and related works. Section 3 presents the proposed DRL approaches for MO optimization in movie recommendation. Section 4 discusses the experimental results. Finally, Section 5 concludes the salient findings of this study.

II. LITERATURE REVIEW

As discussed in the previous section, traditional approaches such as CF, content-based, or even hybrid approaches suffer from cold-start issue and are inadequate to learn autonomously from a dynamic environment [17], [24], [25], [30]. Moreover, these approaches are insufficient to handle MO problems. Most of the general RSs tend to serve users using high-rating or popular items, or similar items in accordance with user's previous preferences in order to achieve high accuracy of prediction. Despite these accuracy metrics being adequate for evaluation purposes, the recommendation quality should not completely rely on accuracy metrics. In other words, other non-accuracy metrics should be considered as well [10]–[13], [16] in order to provide better recommendation quality.

A. MULTI-OBJECTIVE RECOMMENDATION APPROACHES

The existing techniques applied in RS for handling MO optimization problem are mostly EC techniques [31], [36], [56]–[58]. The GA is one of the most popular approaches in EC family that is inspired by biological evolution mechanisms such as nondominated neighbor immune algorithm [33], decomposition-based MO evolutionary algorithm (MOEA/D) [32], and nondominated sorting genetic algorithm II (NSGA-II) [31], [57]. These approaches are not only limited by time-complexity and resource-consuming issues [34] because they require large numbers of iteration and population size for large-scale optimization [49], but they are constrained by premature convergence issue [34]. Furthermore, the EC approaches have difficulty to converge to optimal points [38] in higher-dimensionality MO problem. Moreover, such approaches primarily work on optimization and must be coupled with other prediction algorithms to generate recommendation list. Most of the existing MO studies primarily employed CF technique to predict rating prior to optimization using EC (see Fig. 1) which starts from rating prediction, and generates candidate list, followed by MO optimization.

In the realm of large RS application such as e-commerce platform, the database usually contains vast number of users and at least millions of items that are actively browsed. However, only a small portion of the items are rated by users. Thus, it is impractical to predict the relationship between user and all items since both dimensional quantities increase periodically. Consequently, loads of missing rating values make the user–item matrix scarcely filled. The CF approach that resulted to large data sparsity, also encounters a lot of difficulties to compute similarity between the user or item to identify appropriate items to recommend [2]. The challenge grows even further when it is required to generate recommendation for new users or items added into the system since EC technique is unable to perform optimization without complete item-rating data. The CF method is only learned from the rating matrix, and it is difficult to learn other potential useful features such as user latent.

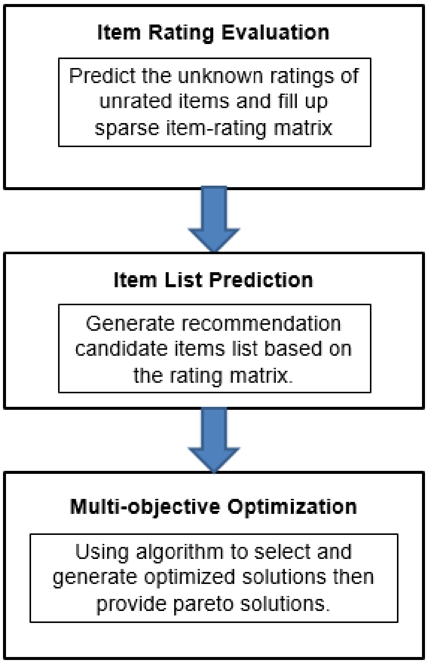
Item Rating Evaluation
Predict the unknown ratings of unrated items and fill up sparse item-rating matrix
Item List Prediction
Generate recommendation candidate items list based on the rating matrix.
Multi-objective Optimization
Using algorithm to select and generate optimized solutions then provide pareto solutions.

FIGURE 1. General process of MO recommendation using CF method combined with EC technique.

In contrast, the nature of DRL in adaptive self-learning through environment exploration and epsilon-greedy policy [47] makes the algorithm itself possess the capacity to sustain in large complex and even sparse environment. With epsilon-greedy policy, DRL agent is allowed to explore the environment for searching better direction and preventing itself from being perplexed at local minima or maxima. The agent has certain policy to perform a random action with probability ε and take action using greedy policy with probability $1 - \varepsilon$. In another words, the ε parameter determines the probability of agent exploration and exploitation as shown in (1).

$$\pi\left(a_t \mid s_t\right) = \begin{cases} 1-\varepsilon & if\ a_t = \arg max_{a_t} Q\left(s_t, a_t\right) \\ \varepsilon & otherwise\ random\ action \end{cases} \quad (1)$$

The epsilon-greedy algorithm ensures all the action space is explored by maintaining a certain exploration probability. Fig. 2 illustrates the interaction between an RL agent with its environment where the agent observes the state from environment and takes action accordingly. Each of the action performed by agent will obtain reward as feedback. Several studies [53], [54], [59]–[64] have demonstrated the robustness of RL algorithm in complex RS application. As suggested by [46], the function approximation algorithm such as DRL approach is more suitable for solving vast state space problem rather than using tabular mapping function such as Q-learning tabular algorithm in numerous state–action environment. One of the most extensively deployed DRL algorithms is DQN [47]. It utilizes deep neural network to approximate Q-values and has been examined in few innovative RS applications [59], [60], [62].

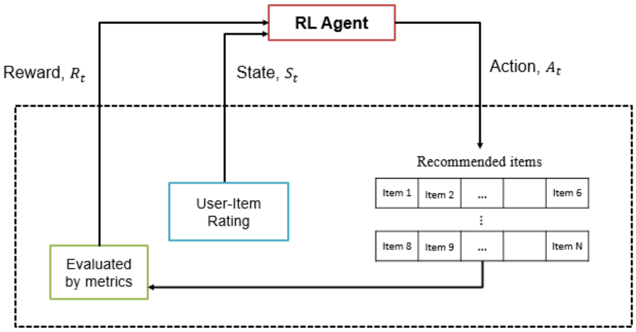

FIGURE 2. Interaction between RL agent with its environment in a typical RL approach.

Inspired by a study [48] that highlighted DQN's advantage in RS by using multi-step user-specific based interactive recommendation with explicit feedback mechanism, we represent the recommendation model as follows in order to adapt RL interaction with the dynamic environment:

- States, ***S***. The state, $s \in \boldsymbol{S}$ is the feature representation that expresses the interaction between a user and a movie; it represents user's behavior along the ascending timestamp.
- Actions, *A*. The *a* is the action executed on a state, *s*. Each $a \in A$ is the is the recommended item list with a fixed length *L* for each user. The item in this case is referred to as a movie, and so the action output is list of movie item denoted by a unique ID number.
- Rewards, *R*. *R(s, a)* is the immediate reward obtained by agent for every action *a* executed in state, *s*. Since the goals of the MO agent is to optimize the evaluation metrics, the reward is summation of metrics values. Higher overall rewards indicate a likely better performance.

A few studies [49]–[51], [65], [66] have demonstrated the robustness of RL algorithms in tackling MO problems (such as power system [38] and traveling salesman problem [54]) and overcome the shortcomings of EC techniques. The stellar performance of DRL methods in MO problem provide strong motivation to this study for solving MORS optimization problem using DQN-based approaches. When dealing with MO issues, there is no perfect solution that can reach the best value in all goals concurrently, and it is inevitable to sacrifice at least one target in order to enhance another.

To further explore the potential of RL in RS, a study that applied Q-learning in single objective RS [67] pointed out that the present ratings are significantly correlated with the sequences of past rating. Hence, this finding is taken as encouragement for this work to further investigate the effect of learning rating in sequence by using LSTM layer. The sequential rating information is referred as the user–item rating data arranged in an ordered sequence, and the position of each entry data is significant. In [67], the entries of the training data had included explicit order of the rating given by user on the movie, which required additional effort to label the sequences of mass data. In this work, the training data is sorted ascendingly on the basis of a timestamp and stacked up to be fed as input to recurrent-based DQN agent without the explicit label of the rating order as the recurrent layer can capture the sorted sequential data directly.

Studies on online RS based on DRL is also available, which exploits the users' responses to the current recommendation results (immediate feedback) to optimize the recommendation strategy. Generative adversarial network (GAN) is used to exploit the users' immediate feedback with Q-learning and actor-critic network [68]. This work also proposed a deep generative adversarial networks-based collaborative filtering approach to optimize the negative sampling method. Despite using different DRL approach, this work is similar to our proposed work in the sense that it uses Q-network. However, since it does not handle MORS, we do not consider it as our benchmark.

Another popular approach in DRL called policy gradient is used in combination with RNN [69] to propose a novel top-N model for long-term prediction in a single RS that focuses on hit-rate (fraction of users for which the correct answer is included in the recommendation list) and Normalized Discounted Cumulative Gain, (NDCG) which measures the quality of ranking. They also proposed a new extended GRU cell named EMGRU, which can efficiently enhance the recommendation accuracy by incorporating additional historical information to address the warm-start scenario in long-term prediction. Another similar work on policy gradient method with dynamic recurrent [70] is built in which a profile constructor with autonomous learning ability is designed to make personalized course recommendation. The approach is proposed to address the exploration-exploitation trade-off issue in constructing user profiles while the recurrent scheme by context-aware learning exploit the user's current knowledge and explore the future preferences.

B. EVALUATION METRICS

Optimization between the competing metrics is obligatory to achieve the optimum values in accuracy and non-accuracy metrics concurrently. As indicated by [31], there is trade-off dilemma between the accuracy and diversity of a recommendation, and it sacrifices the accuracy of one metric to improve the other aspects. The study [33] supported that conflict among matching quality and diversity function requires optimization. On the other hand, the relationship between accuracy and novelty is also competing, as shown in [32]. Hence, both accuracy and non-accuracy metrics are deliberated in the proposed algorithm.

Accuracy, commonly also known as precision metric, is an essential evaluation tool that measures how precise are the prediction results. It is measured as the level of correctness between predicted ratings and actual ratings given by user. It also indicates the proportion of recommended items with a high-rating value in the total user's preferable item list. Several studies [31]–[33], [57], [71] have used the precision function as used in this work to evaluate each recommendation list, as defined in (2).

$$P_r = \frac{L_u \cap T_u}{L} \tag{2}$$

where L_u is the predicted recommendation list that contains items for user u, $L_u = [x_1, x_2, \ldots, x_n]$. T_u is the list of actual items in the test set that the user u rated with high rating. A high-rating item is an item that has been given a rating of 3 or above by the user. L is the length of the recommendation list. The recommendation result will be awarded better precision if the greater number of predicted items appear in the test set's high-rated item list.

On the other hand, the diversity function quantifies the difference between items in the recommendation list. This difference can be described by various topics of items in the recommendation. There are some works [33], [71] that use intra-user diversity to assess the capability of recommending the different items to a user. In [31], the another kind of diversity is proposed, which is based on Shannon's entropy. The measurement in [31] is more comprehensive since it comprised of three principal parts, included topic distribution, number of different topics, and the distribution of a topic for each item in the recommendation list. Hence, the proposed diversity for evaluating the recommendation list is formulated as in (3).

$$D_{L_u} = -\left(\sum\nolimits_{i \in L_u} \frac{\left|t_{x_i}\right|}{\left|z_{L_u}\right|} \cdot \log \frac{\left|t_{x_i}\right|}{\left|z_{L_u}\right|}\right) \cdot Div(L_u) \tag{3}$$

where $Div(L_u)$ is the numbers of topics and its distribution in the recommendation list L_u, $\left|t_{x_i}\right|$ is the amount of topics included in item x_i, and $\left|z_{L_u}\right|$ is the total number of topics in the recommendation list. More precisely, the diversity function is related to the topics of items in the recommendation list.

Novelty denotes the popularity of the recommended items. It is a measure of the ability to recommend low-popularity items to the user, assuming that such items, which are in long tail, are considered novel by the user. According to [31], [57], the novelty function is defined as in (4).

$$N = \frac{1}{M \cdot L} \sum\nolimits_{u=1}^{M} \sum\nolimits_{\alpha \in L_u} \log_2 \left(\frac{M}{N_\alpha}\right) \tag{4}$$

where M is the total number of users and N_α is the number of ratings for item α. The recommended items with lower popularity or fewer ratings received are considered novel and have higher novelty value according to (4).

As demonstrated in the MO optimization research works [31], [32], the accuracy indicator and non-accuracy indicators are contradictory. The existing works often focus only on bi-objective optimization between accuracy and another non-accuracy metrics such as either accuracy against diversity or accuracy against novelty. In order to evaluate the robustness of the proposed method, both diversity and novelty metrics are taken into optimization simultaneously with accuracy as MO problem.

C. MULTI-OBJECTIVE OPTIMIZATION

Generally, MO problem involves more than one constraint, and there is no single or best solution for the problem; instead, it may have several solutions. Therefore, MO problem can be described in mathematically as in (5).

$$max\, f_1(x), f_2(x), \ldots, f_n(x), x \in X \tag{5}$$

where x is solution, n is the number of objective functions, and X is the set of feasible solutions. The purpose of MO optimization is to achieve the optimal solution by trade-off to a certain degree on any objective values, and each objective function is represented by a vector in multi-dimensional space. The MO optimization method is mainly distinguished into scalarization and Pareto methods [72]. The former transforms MO operations into scalar fitness function using the weighted-sum approach as shown in (6).

$$F(x) = w_1 f_1(x) + w_2 f_2(x) + \ldots + w_n f_n(x) \tag{6}$$

where w is the weight assigned to each objective. The optimal solutions of the weighted-sum problem combine all the MO functions into one scalar composite objective function. For Pareto method, the solution vectors represent dominated and non-dominated solutions in objective space. A solution vector x_a that dominates another solution x_b can be defined as (7)

$$\begin{aligned} \forall i &= 1, 2, \ldots, n\, f_i(x_a) \geq f_i(x_b) \\ \wedge \exists j &= 1, 2, \ldots, n\, f_j(x_a) > f_j(x_b) \end{aligned} \tag{7}$$

where if there is no other solution of $f(x_b)$ dominating $f(x_a)$, then x_a is Pareto optimal solution. The dominance solution often requires degradation of one objective function in order to improve the target objective function to achieve optimal value. The non-dominated solution is also referred to as Pareto optimal solution.

Based on literature search, precision, novelty, and diversity are the most common combined metrics optimization in MORS and thus being focused on this study. However, most of the existing works do not provide a Pareto-based solution that combines all these three competing objectives concurrently. In fact, precision, novelty, and diversity metrics are reflecting the essential objectives of higher quality recommender system respectively. Therefore, this indicates a room of improvement for MO RS.

III. DEEP REINFORCEMENT LEARNING-BASED MULTI OBJECTIVE RECOMMENDATION SYSTEM

Two types of DQN approaches and its variant are proposed to adapt in MORS environment to generate recommendation items list for user. The proposed DRL approaches are then compared with benchmark work [31], which applied EC technique coupled with CF method. There are three evaluation metrics that are concurrently taken into optimization, namely precision, novelty, and diversity. As discussed in the previous section, both accuracy and non-accuracy metrics are contradictory to each other such that if one objective is maximized, it degrades the other objective(s).

In this MO optimization work, the effectiveness of weighted-sum strategy and Pareto optimal filtering methods are evaluated. As aligned with the benchmark [31], the length of recommendation list L is fixed at 10 for all users. The recommendation output is evaluated in terms of precision (2), diversity (3), and novelty (4) metrics as proposed by [31]. This work is distinguished from previous studies [31], [33], [57], which only focus on optimization between precision and either novelty or diversity, where we simultaneously study the optimization between three objectives: precision, novelty, and diversity.

A. DQNMORS

When dealing with an enormous state space, utilization of large memory to save all state–action pair values is impracticable and inadequate. Moreover, exploration of every state and updating the Q-values using Q-table would be unrealistic. Therefore, the DQN method that uses function approximator to optimize the policy is more practical. The working principle of the proposed DQN algorithm is aligned with the RL mechanism as illustrated in Fig. 2. The algorithm learns to predict the item for user based on feedback from the interaction with the environment. In the RL algorithm, the agent is considered as a component that make decisions on which item to be recommended. It responsible to act accordance with the observed state from environment, each of the action taken will be rewarded corresponding values as feedback to the agent.

Our proposed algorithm, namely DQNMORS is based on DQN and examined in the recommender environment. Fig. 3 shows the diagram of the proposed DQNMORS with optimization. The DQNMORS architecture consists of experience buffer and two identical networks called predicting network (evaluation) and target network. Both networks are initialized with identical parameters. The Q-function approximator is used to optimize the policy, and the approximator is made up of neural networks that consists of 4 layers including the output layer. The two fully connected hidden layers are connected to output layer for each valid action. The first hidden layer consists of 512 neurons and followed by second hidden layer which consists of 1024 rectified units. Lastly, the output layer made up of 1682 units as there are total 1682 unique movies in dataset. The action performed on respective state and the reward obtained are stored in replay memory for experience replay as tuple form, $e_t = (s_t, a_t, r, s_{t+1})$ at each time step, t. During training, the agent randomly samples the minibatches of transition from the replay memory and then performs gradient descent with respect to the network parameters. The randomly sampling breaks undesirable correlations between the samples and therefore minimize the variance of the updates. The predicting network is updated periodically with parameters from the target network. It is responsible for regulating the action values toward target values, thereby leading to a more stable learning process. The loss function applied in the DRL neural network is mean squared error of predicted Q-value and the target Q-value.

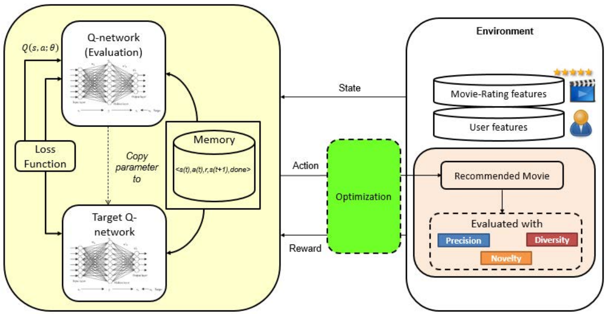

FIGURE 3. Structure of proposed DQNMORS framework.

B. recDQNMORS

he sequence of past ratings information is significantly correlated with the current ratings [67]. This motivates us to investigate further on the impact of learning sequential data by applying recurrent layer to the prediction task. Recurrent neural network (RNN) is mainly used for solving the short-term memory issue in a basic neural network. Few researches [48], [73], [74] pioneered the integrated the RNN with DRL to learn sequential data. In RS domain, [63], [75] employed the hybrid RNN in RL algorithm and demonstrated the ability of capturing long-term sequential information. Thereupon, we fused long-short term memory (LSTM) recurrent layer with the DQNMORS algorithm and named this algorithm as recDQNMORS (see Fig. 4 and Appendix B).

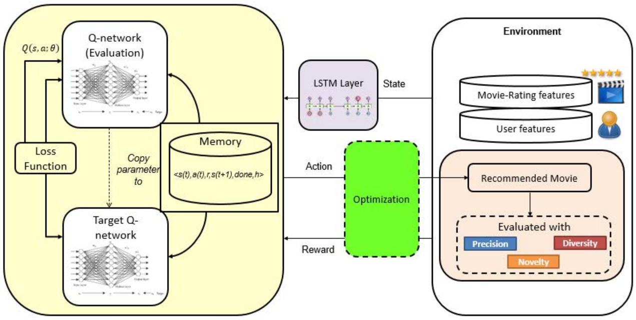

FIGURE 4. Structure of proposed recDQNMORS method.

LSTM [76] is an extension architecture from RNN and is meant to address the short-term memory issue in basic RNN occurring because of vanishing gradient effect. The proposed recDQNMORS approach is modified from LSTM-based recurrent enhanced approach used in [77], which demonstrated the significant role of LSTM in handling data in order. The LSTM is placed on the top layer of network to handle the sequential input data.

C. OPTIMIZATION METHODS

Both scalarization and Pareto method are applied on identical DQNMORS algorithm in order to determine which optimization method has better performance. From the comparison result among DQNMORS approaches, the optimization method that contributes better result is selected to be adopted in recDQNMORS approach for subsequent experiment.

TABLE 1. Summary of algorithm name according to optimization method used with the deep reinforcement learning approaches.

DRL Approach	Algorithm Name	Optimization Method
DQNMORS	DQNMORS_ws	Scalarization method
	DQNMORS_pf	Pareto method

In order to classify conveniently, the name of experimental algorithms is summarized in Table 1.

Both algorithms are applied to identical MO environment but associated with different optimization method. First, the DQNMORS, which used scalarization method, also called weighted-sum method (6) was used to compute the reward value after each recommendation list produced by agent. The reward for this recommended list is an aggregation of each metric functions that have been discussed previously: precision (2), diversity (3), and novelty (4) multiplied with corresponding weights as shown by line 29 in DQNMORS algorithm (Algorithm 1). The weight for each objective is determined in proportion to the relative importance of each objective. Since the importance of each metric is considered equivalent, the weights, w_p, w_d, and $\mathrm{w_n}$, that assigned to each objective respectively is equal to 0.3. The reward function in the proposed DQNMORS_ws framework is established by summation of the evaluation metrics as it reflects directly to DRL agent about the quality of recommended items. On the other hand, the DQNMORS_pf with Pareto method (7) was used to select the optimal recommendation list from the five recommendation lists generated by agent for each individual user. Every recommended list was then evaluated with the metric functions (2), (3), and (4), respectively and only one optimum list was selected as final recommendation list for that user.

IV. EXPERIMENTS SETTING

A. DATASET

To evaluate the performance of the proposed DRL approaches in RS, the agent and environment were designed to be interactive, while a well-known dataset from GroupLens Research, namely MovieLens 100K dataset [78], was utilized. It comprises of 100,000 ratings that scale from 1 to 5, and total 943 individual users with1682 movies. All the users have rated at least 20 movies in the dataset. There are a total of 19 genres, and each movie has devoted to at least 1 genre topic. In this work, the user information was also exploited as latent input for the DRL agent, and the impact of taking the latent user input is discussed in the next section. The user information includes age, gender, occupation, and demographics. These features are related to user personalization and act as a unique representation for every user. In the test set, there are 462 users and only those movie items rated 3 or above in order to accommodate precision evaluation as in [31].

B. INPUT LATENT STATES

The DRL agent interacts with the environment by observing the input state and performs corresponding action (explore or exploit) continuously. The state in RS environment commonly refers to the interaction between the user and the item in the application (according to the dataset). Since the DRL approach is effective in incorporating side features such as user latent into input states, it possesses advantages over other algorithm that incapable to learn the latent features.

A comparison was carried out between DQNMORS with user and without user latent in accordance with user's movie-rating values to validate the effect of incorporating user latent. The comparison was made to justify the impact of user latent on the recommendation result. In order to capture the essential information from the state, the input was embedded as input vector. One-hot encoding is not suitable in this case as it lacks meaningful relations between vectors. Instead, user information was embedded as latent representation. Same setting was applied to recDQNMORS. Table 2 summarizes the differences between the setting of input features. It is hypothesized that the user latent input will benefit the DRL agent since the additional features are strongly related to user personalization, and it is a unique representation for every user in dataset.

TABLE 2. Input features for the proposed deep reinforcement learning agents.

User Feature	Movie-rating Feature
User attributes include age, gender, occupation, zip code, and rated movie ID with corresponding rating value by user.	User ID, movie ID with corresponding rating value by user.

C. SEQUENTIAL INPUT STATES

Since the order of past rating has influence on the present ratings [67], the historical sequential rating data could benefit the agent to generate better predictions. In order to capture sequential input, the LSTM layer is applied on the top of dense neural network. The sequential rating information is referred to as the user–item rating data, which is arranged in ascending order. The recDQNMORS is proposed to study the effect of learning the sequential rating input. In order to verify this assumption, recDQNMORS is compared with DQNMORS by using the same input features and optimization method.

D. HYPERPARAMETER SETTING

Preliminary experiments were conducted to identify the optimum hyperparameter values for DQNMORS and recDQNMORS. The essential hyperparameters such as learning rate, discount factor, or epsilon values could directly control the agents' behavior in learning process. The tuned hyperparameters that were used are encapsulated in Table 3. In general, the hyperparameter-tuning experiments are executed through 30 independent runs in order to collect statistical results. The average metrics values of 10 sample users are taken for analysis and plotted.

E. COMPLEXITY ANALYSIS

The computational complexity of proposed approaches is deduced from its pseudocode as introduced in Algorithm 1 and 2 respectively (Appendix A & B). By referring to the operational rules of the symbol O, the worst-case complexity of the algorithms has been simplified accordingly. The DQNMORS_ws_m_u take a complexity of $O(m^2 \cdot a)$ whereas the DQNMORS_pf_mu and both recDQNMORS algorithms have same complexity of $O(m^2 \cdot P \cdot a)$, where m is the number of users, P is the number of recommendation list, and a is the recommended items by the agent. On the other site, the existing MO optimization algorithm adopt NSGA-II framework, including MOEA-ProbS, PMOEA [31], and MOEA-EPG [58] which possess same computational complexity $O(T \cdot N \cdot m^2 \cdot n)$, where T is the predefined maximum number of generations, N is the population size, and n denotes the number of items. By comparison, the proposed DRL approaches has no higher complexity than benchmark GA.

TABLE 3. Parameter settings for proposed multi-objective deep reinforcement learning agents.

Parameter	DQNMORS_ws_m_u	DQNMORS_pf_m_u	DQNMORS_pf_m	recDQNMORS_pf_m_u	recDQNMORS_pf_m
Input Feature	User Feature	User Feature	Movie-rating	User Feature	Movie-rating
Optimization method	Weighted Sum Method	Pareto Filtering			
Learning Rate	0.0001				
Discount Factor	0.10			0.9	
Epsilon	3.0				
Min. Epsilon	0.5				
Epsilon Decay Rate	0.9	0.7			
Finest Epoch Number	50	10			
Length of the Recommendation list	10				
Number of recommendation list	1	5			

V. RESULTS AND ANALYSIS

Three experiments were conducted to substantiate the proposed algorithms, as summarized in Table 4.

A. SCALARIZATION METHOD VERSUS PARETO METHODS

The results of comparison between the scalarization (weighted-sum) method and Pareto method are presented in

TABLE 4. Summary of experiments.

No	Experiment	Objective
1	Comparison between scalarization method and Pareto method in optimization	To determine which optimization method contributes to better performance
2	Studying the effect of incorporating user feature into input state	To investigate the effect of user latent input on the performance of DRL agent
3	Studying the impact of learning rating data in sequence	To investigate the influence of learning sequential input data

this subsection. As shown in Fig. 5, the DQNMORS_pf_m_u approach with Pareto method outperformed the DQNMORS_ws_m_u approach, which adopts the weighted-sum method, especially in terms of precision and novelty by 13.04% and 2.56%, respectively. The average diversity of DQNMORS_pf_m_u is lower than DQNMORS_ws_m_u by 8.42%.

From the result shown in Fig. 5, DQNMORS_pf_m_u is regarded as a better performer than DQNMORS_ws_m_u from the aspect of average metrics value despite the diversity being slightly lower than DQNMORS_ws_m_u. This indicates that that Pareto method is more effective for attaining higher accuracy while maintaining other non-accuracy metrics. It has better ability to optimize multiple metrics because of Pareto filtering from the solution space without the requirement of assigning weight factor to each objective.

B. USER LATENT FEATURE VERSUS MOVIE-RATING FEATURE

To investigate the impact of user latent feature on the performance, DQNMORS with user latent input (denoted DQNMORS_pf_m_u) is compared against DQNMORS with movie-rating feature input (denoted DQNMORS_pf_m). The average metrics obtained by each agent are shown in Fig. 6. Overall, the results show that the hypothesis is true. i.e., user latent input contributes to better performance. The results from DQNMORS_pf_m_u have surpassed the DQNMORS_pf_m in terms of all evaluation metrics.

DQNMORS_pf_m_u achieved higher precision than DQNMORS_pf_m_u by 19.80% (Fig. 6). According to the average of novelty, the DQNMORS_pf_m_u outperformed DQNMORS_pf_m by 20.46% and attained higher average of diversity by 1.60%. As expected, DQNMORS_pf_m_u, which incorporates user features, can learn more context about the interaction between user and movie item, whereas the DQNMORS_pf_m lacks this information, which affected the performance. Therefore, incorporating user latent as side features beyond the query led to better recommendation results.

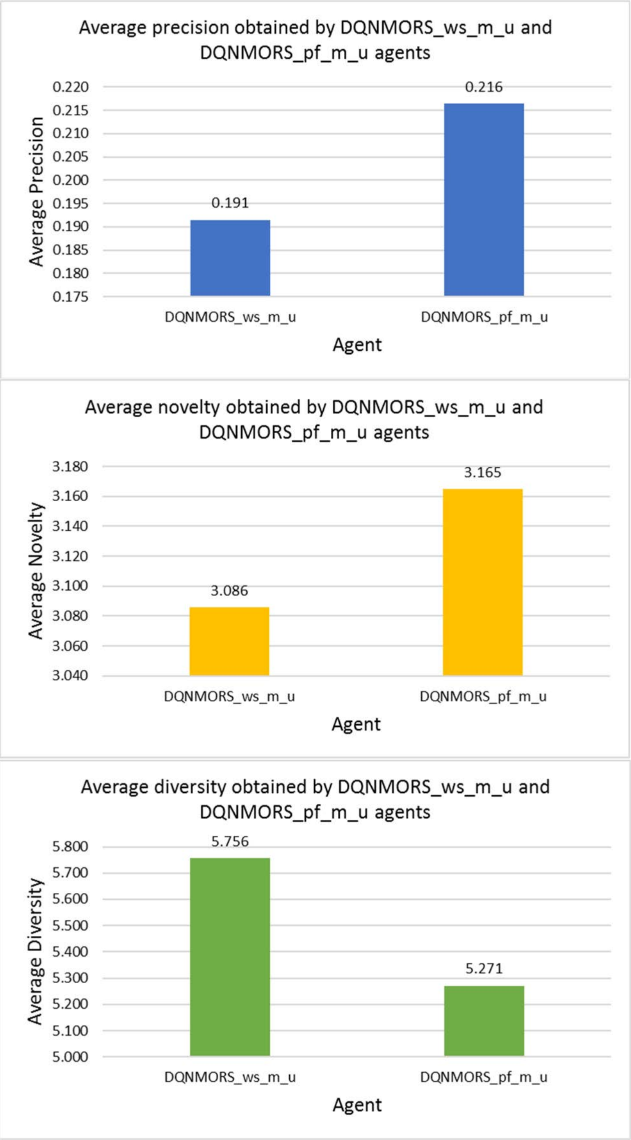

FIGURE 5. Average metrics values of recommendation results obtained by DQNMORS_ws_m_u and DQNMORS_pf_m_u for 10 sample users.

C. LEARNING SEQUENTIAL RATING INPUT

In order to present the impact of learning sequential rating information, the recDQNMORS_pf_m_u algorithm was compared against DQNMORS_pf_m_u algorithm, where both algorithms applied Pareto method for optimization and user latent input sorted in an ascending order according to timestamp. The main difference is only the presence of LSTM layer, where the recDQNMORS algorithm utilizes LSTM layer to capture sequential input states, while the DQNMORS is purely based on DQN without LSTM layer. The comparison results are shown in Fig. 7.

The performance of recDQNMORS_pf_m_u using LSTM layer is not aligned with the expectation. The results show that learning sequential rating data does not enhance the recommendation as the average of precision was unexpectedly lower than DQNMORS_pf_m_u by 17.57% and

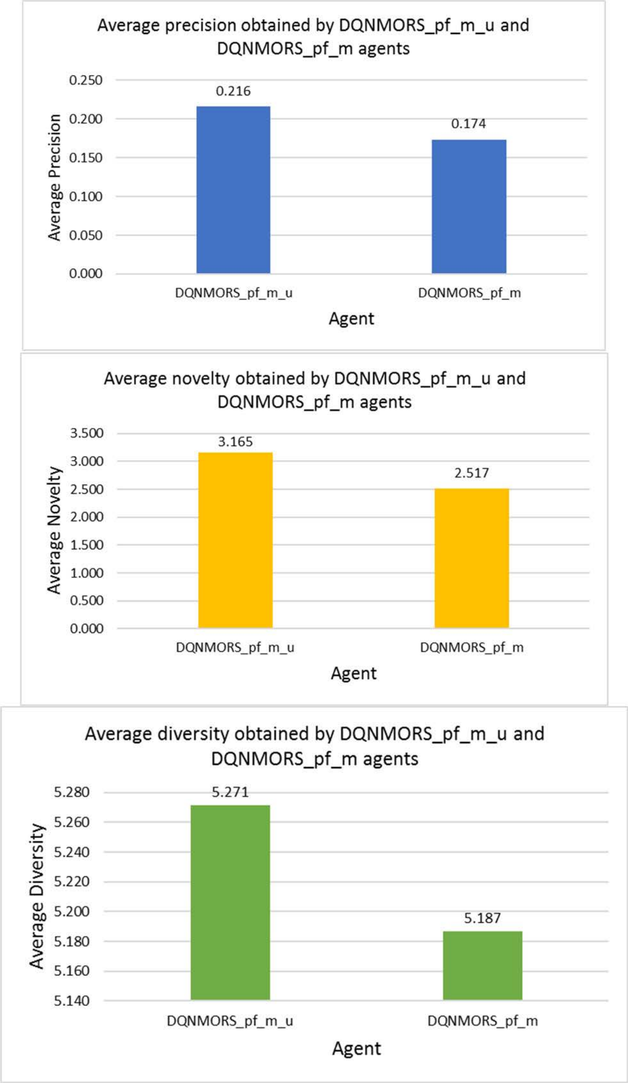

FIGURE 6. Average metric values of recommendation results obtained by DQNMORS_pf_m_u and DQNMORS_pf_m for 10 sample users.

novelty by 4.68%. Only in average diversity, the recDQNMORS_pf_m_u achieved better performance than DQNMORS_pf_m_u by 2.66%. The contradictory performance of recDQNMORS_pf_m_u indicated that learning sequential past ratings has no positive effect to the agent.

The reason behind this result may be the inefficient representation of the sequential rating input state. First, the transition of the user movie-rating across the timestamp actually has no meaningful context to represent the changes of users' preferences. In contrast to the case in [77], which utilized LSTM layer to learn the trend of stock price, the change in stock price provide meaningful signal information to the agent. However, in the case of RS environment, the alteration of user movie-rating did not provide any useful representation. Besides, the MovieLens dataset contained high sparsity on top of watching and rating sequence gaps patterns, and the LSTM layer has difficulty to extract sufficient historical data, thereby causing unstable learning. Therefore, the strength of LSTM layer in this case is not exerted.

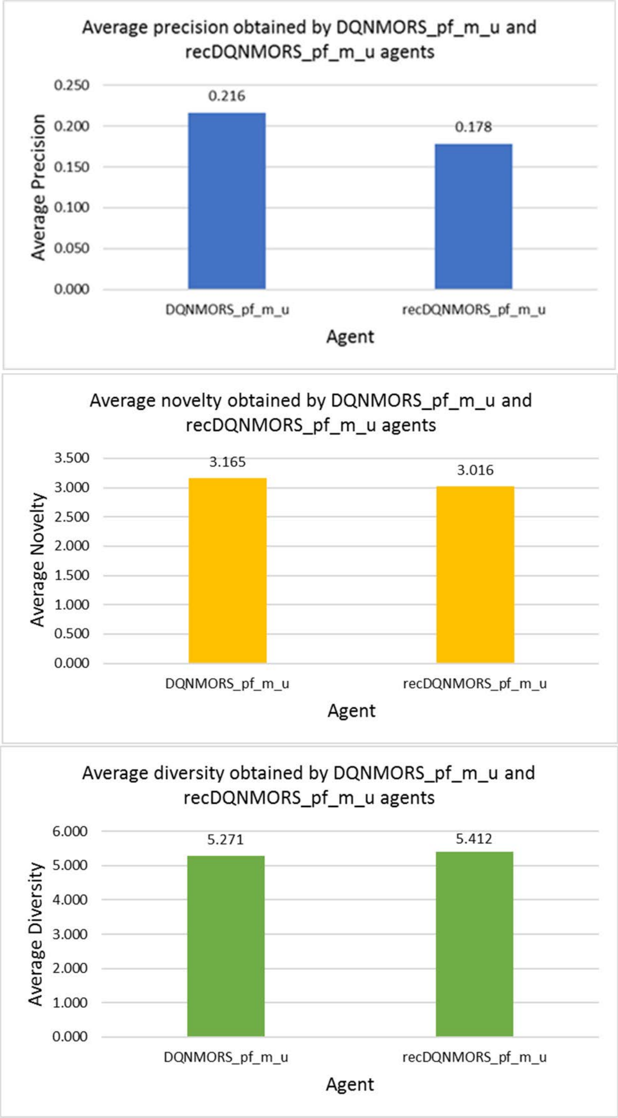

FIGURE 7. Average metrics values of recommendation results obtained by recDQNMORS_pf_m_u approach and recDQNMORS_pf_m_u approach for of 10 sample users.

D. PMOEA VERSUS DQNMORS VERSUS recDQNMORS

The performance of DQNMORS and recDQNMORS algorithms were compared by including comparison against the benchmark results from [31], which utilized probabilistic MO evolutionary algorithm (PMOEA) approaches in terms of precision, novelty, and diversity. The average of mean, minimum, and maximum values of precision, novelty, and diversity metrics from 10 sample users are presented in Appendices C, D, and E.

As evidenced by the result plotted in Fig. 8, the proposed DRL approaches are capable of concurrently handling multiple competing objectives in RS. In general, none of the approach was found to achieve the best results on both metrics simultaneously. The PMOEA+CF_User technique from the benchmark obtained the highest average of mean precision value at 0.50, whereas the highest average of mean precision from DQNMORS_pf_m_u is only 0.22. However, the average of maximum precision values obtained by DQNMORS_ws_m_u and DQNMORS_pf_m_u is still considered competent against the best from benchmark, which achieved 0.50 and 0.46, respectively among the 10 sample users as shown in Appendices C, D, and E. DQNMORS_ws_m_u and DQNMORS_pf_m_u have higher maximum precision than PMOEA+CF_Item for Users 1, 4, and 9, while maximum precision on User 2 and User 8 was achieved with PMOEA+CF_Item.

Although the benchmark PMOEA+CF_User achieved higher precision, it has lower values of novelty and diversity compared to any of the proposed DRL approaches. Both DQNMORS and recDQNMORS have higher average of mean novelty than PMOEA+CF_User, except DQNMORS_pf_m. The recDQNMORS_pf_m achieved higher novelty than PMOEA+CF_Item in all sample users. In average of maximum novelty, all the proposed DRL approaches surpassed all the PMOEA based approaches. In terms of average minimum novelty, the PMOEA+CF_Item has the highest values compared to DRL approaches. However, the majority of PMOEA approaches are considered lower in novelty compared to the proposed DRL approaches.

The exploration–exploitation nature of DRL agents induce higher potential to explore items with more variety, as it enables the agent to reach wider range of items and contribute to better diversity. There is a striking achievement in diversity by all the DQNMORS and recDQNMORS approaches. As shown in the results, both DQNMORS and recDQNMORS approaches surpassed all PMOEA-based approaches in the average of mean diversity. The DQNMORS_ws_m_u and DQNMORS_pf_m_u clearly achieved higher average mean, minimum, and maximum of diversity compared against the PMOEA. The recDQNMORS_pf_m has lowest mean diversity among DRL approaches, but it still outperformed PMOEA+ProbS by 69% in average of mean diversity.

In general, our proposed algorithms have endured in optimization of three constraints simultaneously instead of dual-objectives. Appropriate exploration level support DRL agent to prospecting higher reward action, it advocates agent to discover long-tail items that potentially higher novelty and diversify the categories of recommendation list. The interactions between DRL agent with environment are dictated by a balance of exploration and exploitation, it provides advantages over the GA which vulnerable to premature convergence effect. The premature convergence issue generally is a consequence of losing diversity within the population due to GA operators. The crossover and mutation operator are function for exploitation and exploration respectively by produce genes from available parents. However, it is difficult to generate optimum solutions because of limited items dominated the sub-population and then constraining it to converge to a local optimum. In contrast, the exploration-exploitation strategy in DRL approaches has more flexibility as it provides larger probability for random selection, therefore, it has greater exploration rate to discover more novel diverse items. Compared to GA that applied in [31], it has only fixed probability to randomly select items from the sub-population.

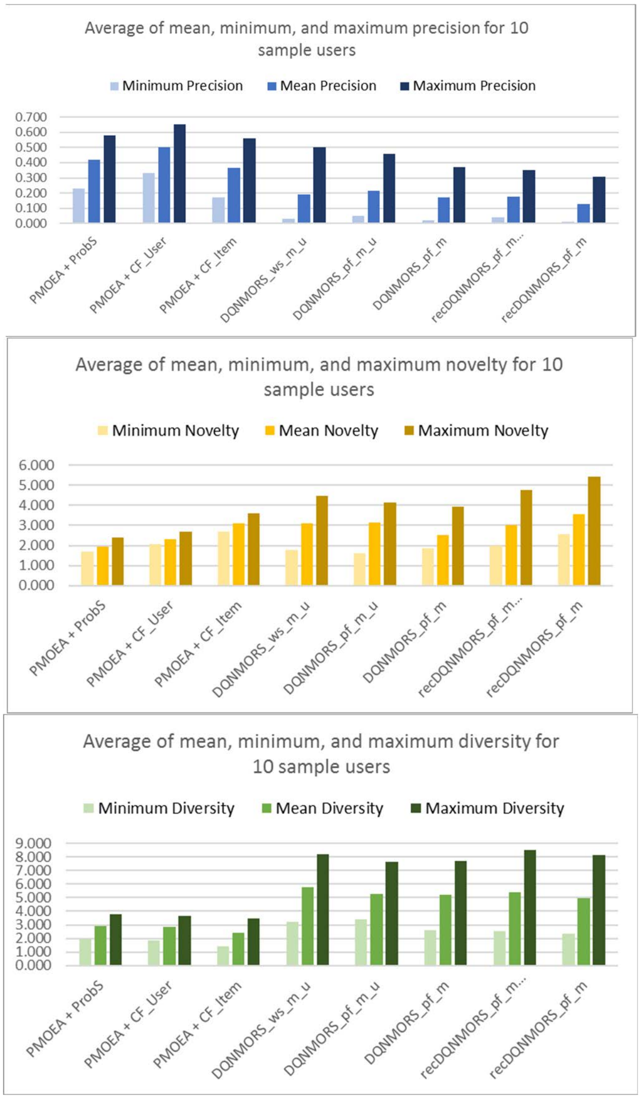

FIGURE 8. **The average of mean, minimum, and maximum metrics values from all the algorithms on 10 sample users.**

The superiority of DRL approach also can be explained by its ability to explicitly represent uncertainty in its transition function and to monitor dynamic changes in the highly sparse environment. The agent able to predict and optimize the recommendation directly without rely on additional separated rating predictor as used in [31]. Although rating prediction tends to secure precision, it disregards novelty and diversity.

Nevertheless, balancing between conflicting objectives required additional efforts, and the cost of considering non-accuracy metrics is that certain degree of precision metrics was required to be abated. As a trade-off, precision of recommendation was affected. The learning performance of DRL agent is heavily rely on reward that obtained after every action taken. However, the reward function for MO problem is always problem dependent, and it is difficult to justify the

Algorithm 1

Input: Set of latent feature input
Output: List of movie items

```
Initialize parameters and replay memory $D$ to capacity $N$
Initialize weights for the evaluation networks, $Q$ and target network $\hat{Q}$
for epoch = 0 to $M$
  for $t$ =1, $T$ in a batch do
     Observe input state, $s_t$ which is latent features
     if using Pareto filtering optimization
        if $\varepsilon >$ random generated number then
              Select random movie items to compose $P$ list of movies with length, $L$ as action, $a_t$
        else
               select the movie items by $a_t = argmax_a(Q(s_t, a))$ to compose $P$ list of movies with length, $L$ as
               action, $a_t$
               for p = 1 to $P$
                  for $a$ = recommendation list, $A$
                     Evaluate each movie list recommendation by precision (2), diversity (3), novelty (4)
                     Use Pareto filtering select the final optimal movie list as $new\_A$ to user
                  end for
               end for
         end if
      else
         if $\varepsilon >$ random generated number then
              Select random movie items with to compose a single movie list with length, $L$ as action, $a_t$
         else
              Select the movie items by $a_t = argmax_a(Q(s_t, a))$ to compose a movie list with length, $L$ as
              action, $a_t$
         end if
      end if
      Set next state, $s_{t+1}$ to current state, $s_t$
      for $a$ = recommendation list, $new\_A$
         Evaluate each movie list recommendation by precision (2), diversity (3), novelty (4)
         if using weighted sum technique as optimization then
            $r = (w_p \times precision + w_d \times diversity + w_n \times novelty)$
         else
            $r = precision + diversity + novelty$
         end if
         Store experience $(s_t, a_t, r, s_{t+1})$ in D
      end for
      Sample random minibatch of transition from memory

            $\text{Set} y_j = \begin{cases} r_j & \text{if the next state user is new user} \\ r_j + \gamma \max \hat{Q}\left(\phi_{j+1}, a_{j+1}; \theta^-\right) & \text{otherwise} \end{cases}$

      Perform gradient descent step on $(y_j - Q(\phi_j, a_j; \theta))^2$ w.r.t parameter $\theta$ in network
      Copy weight parameter from evaluation Q-network into target network, $\hat{Q}$
      if $\varepsilon > \varepsilon_{min}$ then
           $\varepsilon$ multiply with $\varepsilon$-decay rate
      end if
    end for
  end for
```

Algorithm 2

```
Input: Group of latent feature input arranged according to timestamp.
Output: List of movie items

Initialize parameters and replay memory $D$ to capacity $N$
Initialize weights for the Q-networks, $Q$ and target network $\hat{Q}$
for epoch = 0 to $M$
   for $t$ =1, $T$ in a batch do
      Observe input state, $s_t$ which is latent features
      if $\varepsilon >$ random generated number then
         Select random movie items with probability $\varepsilon$ to compose $P$ list of movies with length, $L$ as action, $a_t$
      else
         Select the movie items by $a_t = argmax_a(Q(s_t, a))$ to compose $P$ list of movies with length, $L$ as action, $a_t$
      end if
      Set next state, $s_{t+1}$ to current state, $s_t$
      for p = 1 to $P$
         for $a$ = recommendation list, $A$
            Evaluate each movie list recommendation by precision (2), diversity (3), novelty (4)
            Use Pareto filtering select the optimal movie list to user
            $r = precision + diversity + novelty$
            Store experience $(s_t, a_t, r, s_{t+1}, h)$ in D
         end for
      end for
      Sample random minibatch of transition from memory
      Copy weight parameter from Q-network into target network, $\hat{Q}$
         Set $y_j = \begin{cases} r_j & \text{if the next state user is new user} \\ r_j + \gamma \max \hat{Q}\left(\phi_{j+1}, a_{j+1}; \theta^-\right) & \text{otherwise} \end{cases}$
      Perform gradient descent step on $(y_j - Q(\phi_j, a_j; \theta))^2$ w.r.t parameter $\theta$ in network
      if $\varepsilon > \varepsilon_{min}$ then
         $\varepsilon$ multiply with $\varepsilon$-decay rate
      end if
   end for
end for
```

effectiveness of the reward function. This limitation is also exhibited by GA as the designing process of fitness function is daunting. Besides that, the learning rate hyperparameter value is set to static along the training which may lead to longer time to converge, however, larger learning rate will cause dramatic effect of learning a sub-optimal set of weights. Hence, dynamic learning rate should be considered.

VI. CONCLUSION

This work presented two DRL approaches, DQNMORS and recDQNMORS, capable of tackling MO problem in RS environment. These algorithms are proposed to optimize three different objectives or metrics, which are precision, novelty, and diversity. From the comparison of optimization methods, the Pareto method was observed to outperform the scalarization method. The DQNMORS approach was further investigated by incorporating user latent features as side feature, and the results show that the additional feature input improved the recommendation performance. Furthermore, DQNMORS was appended with LSTM layer and transformed to recDQNMORS for dealing with learning sequential input data, which regular neural network has difficulty to capture. Although recDQNMORS results have not achieved better precision than DQNMORS owing to ineffective input representation for agent, the ability of optimization is exhibited, and it achieved better result in terms of novelty and diversity.

As for future direction, more advanced DRL approaches can be investigated in terms of robustness and complexity. For instance, multiple networks DRL approach such as Actor–Critic has the potential to increase efficiency since it has double networks to learn value and policy functions. This work sets a benchmark for DRL-based approach in RS application for future research in this topic. Optimizing more than one objective concurrently will endure at least one objective, and it is still a main challenge. Lastly, the sequential rating input required a better technique to capture significant

TABLE 5. Result for average mean, minimum, and maximum precision of 10 sample users by all the proposed algorithms and the benchmark results.

User	PMOEA + ProbS			PMOEA + CF_User			PMOEA + CF_Item			DQNMORS_ws_m_u			DQNMORS_pf_m_u			DQNMORS_pf_m			recDQNMORS_pf_m_u			recDQNMORS_pf_m		
	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max
1	0.690	0.500	0.900	0.767	0.600	1.000	0.556	0.200	0.800	0.463	0.200	0.900	0.607	0.300	0.900	0.380	0.200	0.600	0.437	0.100	0.700	0.370	0.000	0.600
2	0.235	0.000	0.300	0.246	0.000	0.400	0.207	0.000	0.400	0.119	0.000	0.400	0.117	0.000	0.400	0.048	0.000	0.200	0.037	0.000	0.200	0.030	0.000	0.200
3	0.173	0.000	0.300	0.211	0.100	0.400	0.216	0.000	0.400	0.041	0.000	0.300	0.017	0.000	0.100	0.008	0.000	0.100	0.000	0.000	0.000	0.000	0.000	0.000
4	0.273	0.100	0.400	0.155	0.000	0.200	0.023	0.000	0.100	0.048	0.000	0.300	0.071	0.000	0.300	0.048	0.000	0.100	0.021	0.000	0.100	0.010	0.000	0.100
5	0.481	0.300	0.700	0.634	0.400	0.900	0.587	0.300	0.800	0.163	0.000	0.600	0.122	0.000	0.500	0.140	0.000	0.300	0.158	0.000	0.400	0.170	0.100	0.400
6	0.545	0.300	0.800	0.579	0.400	0.700	0.491	0.300	0.800	0.293	0.000	0.700	0.272	0.100	0.500	0.272	0.000	0.500	0.289	0.000	0.600	0.200	0.000	0.500
7	0.799	0.600	1.000	0.912	0.800	1.000	0.653	0.400	0.900	0.385	0.000	0.700	0.311	0.000	0.700	0.352	0.000	0.700	0.437	0.200	0.700	0.240	0.000	0.400
8	0.509	0.300	0.700	0.600	0.400	0.800	0.290	0.100	0.500	0.178	0.000	0.500	0.333	0.000	0.500	0.172	0.000	0.600	0.153	0.000	0.400	0.080	0.000	0.300
9	0.081	0.000	0.100	0.071	0.000	0.100	0.000	0.000	0.000	0.004	0.000	0.100	0.010	0.000	0.100	0.008	0.000	0.100	0.000	0.000	0.000	0.020	0.000	0.200
10	0.397	0.200	0.600	0.822	0.600	1.000	0.658	0.400	0.900	0.222	0.100	0.500	0.305	0.100	0.600	0.308	0.000	0.500	0.253	0.100	0.400	0.140	0.000	0.400
Average	0.418	0.230	0.580	0.500	0.330	0.650	0.368	0.170	0.560	0.191	0.030	0.500	0.216	0.050	0.460	0.174	0.020	0.370	0.178	0.040	0.350	0.126	0.010	0.310

TABLE 6. Result for average mean, minimum, and maximum novelty of 10 sample users by all the proposed algorithms and the benchmark results.

User	PMOEA + ProbS			PMOEA + CF_User			PMOEA + CF_Item			DQNMORS_ws_m_u			DQNMORS_pf_m_u			DQNMORS_pf_m			recDQNMORS_pf_m_u			recDQNMORS_pf_m		
	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max
1	1.915	1.684	2.252	2.087	1.795	2.394	2.618	2.292	3.240	2.968	1.684	4.274	3.561	1.716	3.743	2.535	1.815	3.700	2.678	2.027	3.515	3.607	2.512	5.096
2	1.989	1.761	2.411	2.616	2.236	2.973	3.193	2.812	3.679	3.372	1.826	5.713	3.206	1.497	5.161	2.715	1.845	4.595	2.959	1.734	3.922	3.440	2.512	5.096
3	2.100	1.838	2.675	2.587	2.259	2.934	3.483	3.081	4.014	3.161	1.695	4.946	3.541	1.474	3.835	2.472	1.923	3.717	3.172	2.053	5.636	3.663	2.512	6.155
4	2.111	1.891	2.441	2.424	2.151	3.038	3.175	2.631	4.008	3.028	1.743	4.274	2.688	1.663	4.000	2.427	1.869	3.571	3.114	2.054	4.219	3.660	2.784	5.571
5	1.873	1.669	2.148	2.210	1.981	2.459	2.759	2.341	3.188	3.194	1.733	5.255	2.824	1.587	4.218	2.466	1.818	4.131	3.396	2.208	4.862	3.398	2.512	4.896
6	1.943	1.672	2.289	2.347	2.114	2.751	3.001	2.410	3.511	3.103	2.002	4.274	3.578	1.672	3.639	2.561	1.815	3.937	2.804	1.822	5.854	3.562	2.512	5.132
7	1.999	1.700	2.525	2.208	1.985	2.669	3.503	3.151	3.852	3.044	1.760	4.135	3.199	1.808	3.901	2.482	2.039	4.104	2.922	2.018	4.174	3.643	2.512	6.156
8	1.956	1.576	2.739	2.304	2.032	2.648	3.242	2.727	3.754	2.965	1.759	3.896	2.717	1.646	4.880	2.492	1.815	3.833	2.981	2.016	3.922	3.460	2.694	5.096
9	1.778	1.499	2.199	1.780	1.633	2.038	2.489	2.126	2.899	2.997	1.600	3.851	3.502	1.511	3.773	2.515	1.957	3.682	3.119	2.056	5.636	3.447	2.512	4.896
10	1.901	1.601	2.177	2.509	2.324	2.907	3.581	3.288	3.873	3.023	1.875	4.274	2.830	1.575	4.443	2.507	1.815	4.006	3.017	2.027	6.097	3.698	2.512	6.121
Average	1.957	1.689	2.386	2.307	2.051	2.681	3.104	2.686	3.602	3.086	1.768	4.489	3.165	1.615	4.159	2.517	1.871	3.927	3.016	2.002	4.784	3.558	2.558	5.422

latent information in order to enhance sequential decision-making.

APPENDICES

Appendices A and B present the algorithms of the proposed DQNMORS and recDQNMORS, respectively, followed by experiment results of all proposed algorithms against benchmark in Appendices C, D, and E.

APPENDIX A DQNMORS ALGORITHM

See Algorithm 1.

APPENDIX B recDQNMORS ALGORITHM

See Algorithm 2.

TABLE 7. Result for average mean, minimum, and maximum diversity of 10 sample users by all the proposed algorithms and the benchmark results.

User	PMOEA + ProbS			PMOEA + CF_User			PMOEA + CF_Item			DQNMORS_ws_m_u			DQNMORS_pf_m_u			DQNMORS_pf_m			recDQNMORS_pf_m_u			recDQNMORS_pf_m		
	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max
1	2.531	1.797	3.809	3.098	2.090	4.232	3.200	1.828	4.178	5.842	3.151	7.638	5.033	3.222	6.819	4.798	2.899	6.840	5.371	3.036	7.828	4.559	1.650	8.542
2	2.256	1.786	2.848	2.016	1.108	2.873	1.347	0.510	2.419	5.815	2.933	8.243	4.785	2.888	6.984	5.643	2.858	8.127	5.352	2.708	10.013	5.139	3.016	6.974
3	3.307	2.177	4.494	2.028	1.358	2.545	1.852	1.031	2.846	5.598	3.105	8.586	5.432	3.666	8.586	5.141	1.513	7.522	5.264	1.526	8.552	4.794	1.041	7.365
4	3.175	1.519	4.048	2.805	2.102	3.622	3.256	2.429	4.126	5.991	2.106	8.510	5.013	3.138	6.871	5.070	3.110	8.503	5.672	3.759	8.737	5.354	3.694	8.363
5	2.903	1.806	3.770	2.987	1.797	4.528	3.011	2.133	4.164	5.799	2.923	7.640	5.052	3.399	7.011	5.159	2.330	8.182	5.263	2.738	6.977	5.256	3.016	8.293
6	3.002	2.125	3.642	3.056	2.051	4.551	2.657	1.427	3.823	5.660	4.147	8.345	5.194	3.201	7.665	5.162	3.583	7.732	5.402	2.706	7.811	4.791	1.062	8.235
7	2.765	1.779	3.532	2.916	2.151	3.568	1.906	0.666	3.647	5.761	3.750	8.632	5.699	3.813	9.451	5.227	1.643	7.616	5.505	3.166	8.449	4.879	1.113	8.372
8	3.055	2.184	3.754	3.053	1.735	2.032	2.253	1.414	3.110	5.523	2.510	8.546	5.605	3.473	7.679	4.844	2.746	6.888	5.613	3.036	8.681	5.196	3.016	8.124
9	3.399	2.542	4.240	2.917	1.796	3.784	2.667	1.529	3.656	5.747	3.805	8.345	5.433	3.847	7.756	5.626	2.579	7.723	5.216	1.526	7.776	4.718	3.016	8.234
10	2.857	1.724	3.703	3.450	2.597	4.645	2.117	1.352	2.964	5.823	3.886	7.799	5.466	3.675	7.638	5.198	2.760	7.647	5.457	1.362	10.221	4.908	3.016	8.726
Average	2.925	1.944	3.784	2.833	1.879	3.638	2.427	1.432	3.493	5.756	3.232	8.229	5.271	3.432	7.646	5.187	2.602	7.678	5.412	2.556	8.504	4.959	2.364	8.123

APPENDIX C

See Table 5.

APPENDIX D

See Table 6.

APPENDIX E

See Table 7.

REFERENCES

[1] Y. Zheng and D. Wang, "A survey of recommender systems with multi-objective optimization," *Neurocomputing*, vol. 474, pp. 141–153, Feb. 2022, doi: 10.1016/j.neucom.2021.11.041.

[2] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, "GroupLens: Applying collaborative filtering to usenet news," *Commun. ACM*, vol. 40, no. 3, pp. 77–87, Mar. 1997, doi: 10.1145/245108.245126.

[3] P. Resnick, P. Bergstrom, and J. Riedl, "GroupLens: An open architecture for collaborative filtering of netnews," in *Proc. ACM Conf. Comput. Supported Cooperat. Work*, 1994, pp. 175–186.

[4] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Commun. ACM*, vol. 35, no. 12, pp. 61–70, 1992, doi: 10.1145/138859.138867.

[5] J. Verhoeff, W. Goffman, and J. Belzer, "Inefficiency of the use of Boolean functions for information retrieval systems," *Commun. ACM*, vol. 4, no. 12, pp. 557–558, Dec. 1961, doi: 10.1145/366853.366861.

[6] F. Park, "Using social recommendation and content-based as classification: Information in recommendation," *Amer. Assoc. Artif. Intell.*, 1998, pp. 714–720.

[7] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009, doi: 10.1109/MC.2009.263.

[8] B. Magnini and C. Strapparava, "Improving user modelling with content-based techniques," in *Proc. Int. Conf. User Modeling*, vol. 2109, 2001, pp. 74–83, doi: 10.1007/3-540-44566-8_8.

[9] A. Pawlicka, M. Pawlicki, R. Kozik, and R. S. Choraś, "A systematic review of recommender systems and their applications in cybersecurity," *Sensors*, vol. 21, no. 15, pp. 1–25, 2021, doi: 10.3390/s21155248.

[10] N. Hurley and M. Zhang, "Novelty and diversity in top-N recommendation—Analysis and evaluation," *ACM Trans. Internet Technol.*, vol. 10, no. 4, pp. 1–30, Mar. 2011, doi: 10.1145/1944339.1944341.

[11] E. M. Hamedani and M. Kaedi, "Recommending the long tail items through personalized diversification," *Knowl.-Based Syst.*, vol. 164, pp. 348–357, Jan. 2019, doi: 10.1016/j.knosys.2018.11.004.

[12] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowl. Syst.*, vol. 46, pp. 109–132, Jul. 2013, doi: 10.1016/j.knosys.2013.03.012.

[13] T. Zhou, X. Zhu, H. Tian, X. Ren, B. Cai, and Y. Huang, "Accurate and diverse recommendations via eliminating redundant correlations," *New J. Phys.*, vol. 11, no. 12, 2009, Art. no. 123008, doi: 10.1088/1367-2630/11/12/123008.

[14] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversification," in *Proc. 14th Int. Conf. World Wide Web (WWW)*, 2005, pp. 22–32.

[15] P. Castells, N. J. Hurley, and S. Vargas, "Novelty and diversity in recommender systems," in *Recommender Systems Handbook*. Boston, MA, USA: Springer, 2015.

[16] S. M. McNee, J. Riedl, and J. A. Konstan, "Being accurate is not enough: How accuracy metrics have hurt recommender systems," in *Proc. CHI Extended Abstr. Hum. Factors Comput. Syst.*, Apr. 2006, pp. 1097–1101.

[17] P. Aggarwal, V. Tomar, and A. Kathuria, "Comparing content based and collaborative filtering in recommender systems," *Int. J. New Technol. Res.*, vol. 3, no. 4, pp. 65–67, 2017.

[18] T. Hyup, K. Joo, and I. Han, "The collaborative filtering recommendation based on SOM cluster-indexing CBR," *Expert Syst. Appl.*, vol. 25, pp. 413–423, Oct. 2003, doi: 10.1016/S0957-4174(03)00067-8.

[19] J. Golbeck and U. Kuter, "Trust metrics in recommender systems," in *Computing With Social Trust*. London, U.K.: Springer, 2009, pp. 169–181, doi: 10.1007/978-1-84800-356-9.

[20] N. Manouselis, K. Verbert, H. Drachsler, and O. C. Santos, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," in *Proc. 4th ACM Conf. Rec. Syst. (RecSys)*, 2010, vol. 17, no. 6, p. 377, doi: 10.1145/1864708.1864797.

[21] I. A. A. Q. Al-Hadi, N. M. Sharef, M. N. Sulaiman, N. Mustapha, and M. Nilashi, "Latent based temporal optimization approach for improving the performance of collaborative filtering," *PeerJ Comput. Sci.*, vol. 6, pp. 1–25, Dec. 2020, doi: 10.7717/PEERJ-CS.331.

[22] A. Zahid, N. M. Sharef, and A. Mustapha, "Normalization-based neighborhood model for cold start problem in recommendation system," *Int. Arab J. Inf. Technol.*, vol. 17, no. 3, pp. 281–290, 2020, doi: 10.34028/iajit/17/3/1.

[23] M. Nilashi, A. Ahani, M. D. Esfahani, E. Yadegaridehkordi, S. Samad, O. Ibrahim, N. M. Sharef, and E. Akbari, "Preference learning for eco-friendly hotels recommendation: A multi-criteria collaborative filtering approach," *J. Cleaner Prod.*, vol. 215, pp. 767–783, Apr. 2019, doi: 10.1016/j.jclepro.2019.01.012.

[24] T. K. Balaji, C. S. R. Annavarapu, and A. Bablani, "Machine learning algorithms for social media analysis: A survey," *Comput. Sci. Rev.*, vol. 40, May 2021, Art. no. 100395, doi: 10.1016/j.cosrev.2021.100395.

[25] B. Sanghavi, R. Rathod, and D. Mistry, "Recommender systems—Comparison of content-based filtering and collaborative filtering," *Int. J. Current Eng. Technol.*, vol. 4, no. 5, pp. 3131–3133, 2014.

[26] K. Soni, R. Goyal, B. Vadera, and S. More, "A three way hybrid movie recommendation syste," *Int. J. Comput. Appl.*, vol. 160, no. 9, pp. 29–32, Feb. 2017.

[27] K. Jung, D. Park, and J. Lee, "Hybrid collaborative filtering and content-based filtering for improved recommender system," in *Proc. Int. Conf. Comput. Sci.*, vol. 3036, 2004, pp. 295–302, doi: 10.1007/978-3-540-24685-5_37.

[28] X. Song, Y. Guo, Y. Chang, F. Zhang, J. Tan, J. Yang, and X. Shi, "A hybrid recommendation system for marine science observation data based on content and literature filtering," *Sensors*, vol. 20, no. 22, p. 6414, Nov. 2020, doi: 10.3390/s20226414.

[29] F. S. Gohari and M. J. Tarokh, "Classification and comparison of the hybrid collaborative filtering systems," *Int. J. Res. Ind. Eng.*, vol. 6, no. 2, pp. 129–148, 2017.

[30] S. Nadi, M. H. Saraee, and A. Bagheri, "A hybrid recommender system for dynamic web users," *Int. J. Multimedia Image Process.*, vol. 1, nos. 1–2, pp. 3–8, Mar. 2011.

[31] L. Cui, P. Ou, X. Fu, Z. Wen, and N. Lu, "A novel multi-objective evolutionary algorithm for recommendation systems," *J. Parallel Distrib. Comput.*, vol. 103, pp. 53–63, May 2017, doi: 10.1016/j.jpdc.2016.10.014.

[32] S. Wang, M. Gong, H. Li, and J. Yang, "Multi-objective optimization for long tail recommendation," *Knowl.-Based Syst.*, vol. 104, pp. 145–155, Jul. 2016, doi: 10.1016/j.knosys.2016.04.018.

[33] B. Geng, L. Li, L. Jiao, M. Gong, Q. Cai, and Y. Wu, "NNIA-RS: A multi-objective optimization based recommender system," *Phys. A, Stat. Mech. Appl.*, vol. 424, pp. 383–397, Apr. 2015, doi: 10.1016/j.physa.2015.01.007.

[34] T. Horváth and A. C. P. L. F. de Carvalho, "Evolutionary computing in recommender systems: A review of recent research," *Natural Comput.*, vol. 16, no. 3, pp. 441–462, Sep. 2017, doi: 10.1007/s11047-016-9540-y.

[35] Q. Zhang, J. Lu, and Y. Jin, "Artificial intelligence in recommender systems," *Complex Intell. Syst.*, vol. 7, no. 1, pp. 439–457, Feb. 2021, doi: 10.1007/s40747-020-00212-w.

[36] A. Abbas, L. Zhang, and S. U. Khan, "A survey on context-aware recommender systems based on computational intelligence techniques," *Computing*, vol. 97, no. 7, pp. 667–690, Jul. 2015, doi: 10.1007/s00607-015-0448-7.

[37] Z. Wang and A. Sobey, "A comparative review between genetic algorithm use in composite optimisation and the state-of-the-art in evolutionary computation," *Compos. Struct.*, vol. 233, Feb. 2020, Art. no. 111739.

[38] M. M. Drugan, "Reinforcement learning versus evolutionary computation: A survey on hybrid algorithms," *Swarm Evol. Comput.*, vol. 44, pp. 228–246, Feb. 2019, doi: 10.1016/j.swevo.2018.03.011.

[39] R. S. Fortes, D. X. de Sousa, D. G. Coelho, A. M. Lacerda, and M. A. Gonçalves, "Individualized extreme dominance (IndED): A new preference-based method for multi-objective recommender systems," *Inf. Sci.*, vol. 572, pp. 558–573, Sep. 2021, doi: 10.1016/j.ins.2021.05.037.

[40] Q. Shambour, "A deep learning based algorithm for multi-criteria recommender systems," *Knowl.-Based Syst.*, vol. 211, Jan. 2021, Art. no. 106545, doi: 10.1016/j.knosys.2020.106545.

[41] M. Hong and J. J. Jung, "Multi-criteria tensor model for tourism recommender systems," *Expert Syst. Appl.*, vol. 170, May 2021, Art. no. 114537, doi: 10.1016/j.eswa.2020.114537.

[42] C. Watkins, "Learning from delayed rewards," Ph.D. thesis, King's College, Cambridge, U.K., 1989.

[43] M. B. Naghibi-Sistani, M. R. Akbarzadeh-Tootoonchi, M. H. J.-D. Bayaz, and H. Rajabi-Mashhadi, "Application of Q-learning with temperature variation for bidding strategies in market based power systems," *Energy Convers. Manage.*, vol. 47, nos. 11–12, pp. 1529–1538, Jul. 2006, doi: 10.1016/j.enconman.2005.08.012.

[44] F. Li, K.-Y. Lam, Z. Sheng, X. Zhang, K. Zhao, and L. Wang, "Q-learning-based dynamic spectrum access in cognitive industrial Internet of Things," *Mobile Netw. Appl.*, vol. 23, no. 6, pp. 1636–1644, Dec. 2018.

[45] T. Mahmood, G. Mujtaba, and A. Venturini, "Dynamic personalization in conversational recommender systems," *Inf. Syst. e-Bus. Manage.*, vol. 12, no. 2, pp. 213–238, May 2014, doi: 10.1007/s10257-013-0222-3.

[46] X. Tang, Y. Chen, X. Li, J. Liu, and Z. Ying, "A reinforcement learning approach to personalized learning recommendation systems," *Brit. J. Math. Stat. Psychol.*, vol. 72, no. 1, pp. 108–135, Feb. 2019, doi: 10.1111/bmsp.12144.

[47] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Feb. 2015, doi: 10.1038/nature14236.

[48] C. Chen, V. Ying, and D. Laird. (2016). *Deep Q-Learning With Recurrent Neural Networks*. [Online]. Available: http://cs229.stanford.edu/proj2016/report/ChenYingLaird-DeepQLearningWithRecurrentNeuralNetwords-report.pdf.

[49] K. Li, T. Zhang, and R. Wang, "Deep reinforcement learning for multiobjective optimization," *IEEE Trans. Cybern.*, vol. 51, no. 6, pp. 3103–3114, Jun. 2021, doi: 10.1109/TCYB.2020.2977661.

[50] H. L. Liao, Q. H. Wu, and L. Jiang, "Multi-objective optimization by reinforcement learning for power system dispatch and voltage stability," in *Proc. IEEE PES Innov. Smart Grid Technol. Conf. Eur. (ISGT Europe)*, Oct. 2010, pp. 1–8, doi: 10.1109/ISGTEUROPE.2010.5638914.

[51] Y. Wang, H. Liu, W. Zheng, Y. Xia, Y. Li, P. Chen, K. Guo, and H. Xie, "Multi-objective workflow scheduling with deep-Q-network-based multi-agent reinforcement learning," *IEEE Access*, vol. 7, pp. 39974–39982, 2019, doi: 10.1109/ACCESS.2019.2902846.

[52] F. Liu, R. Tang, X. Li, W. Zhang, Y. Ye, H. Chen, H. Guo, and Y. Zhang, "Deep reinforcement learning based recommendation with explicit user-item interactions modeling," 2018, *arXiv:1810.12027*.

[53] I. Munemasa, Y. Tomomatsu, K. Hayashi, and T. Takagi, "Deep reinforcement learning for recommender systems," in *Proc. Int. Conf. Inf. Commun. Technol. (ICOIACT)*, Mar. 2018, pp. 226–233.

[54] S.-Y. Chen, Y. Yu, Q. Da, J. Tan, H.-K. Huang, and H.-H. Tang, "Stabilizing reinforcement learning in dynamic environment with application to online recommendation," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 1187–1196, doi: 10.1145/3219819.3220122.

[55] G. Zheng, F. Zhang, Z. Zheng, Y. Xiang, N. J. Yuan, X. Xie, and Z. Li, "DRN: A deep reinforcement learning framework for news recommendation," in *Proc. World Wide Web Conf. (WWW)*, Lyon, France, 2018, pp. 167–176, doi: 10.1145/3178876.3185994.

[56] W. Ma, X. Feng, S. Wang, and M. Gong, "Personalized recommendation based on heat bidirectional transfer," *Phys. A, Stat. Mech. Appl.*, vol. 444, pp. 713–721, Feb. 2016, doi: 10.1016/j.physa.2015.10.068.

[57] Y. Zuo, M. Gong, J. Zeng, L. Ma, and L. Jiao, "Personalized recommendation based on evolutionary multi-objective optimization," *IEEE Comput. Intell. Mag.*, vol. 10, no. 1, pp. 52–62, Feb. 2015, doi: 10.1109/MCI.2014.2369894.

[58] Q. Lin, X. Wang, B. Hu, L. Ma, F. Chen, J. Li, and C. A. C. Coello, "Multiobjective personalized recommendation algorithm using extreme point guided evolutionary computation," *Complexity*, vol. 2018, pp. 1–18, Nov. 2018, doi: 10.1155/2018/1716352.

[59] Y. Lei and W. Li, "Interactive recommendation with user-specific deep reinforcement learning," *ACM Trans. Knowl. Discovery From Data*, vol. 13, no. 6, pp. 1–15, Dec. 2019.

[60] C. Tan, R. Han, R. Ye, and K. Chen, "Adaptive learning recommendation strategy based on deep Q-learning," *Appl. Psychol. Meas.*, vol. 44, no. 4, pp. 1–16, 2019, doi: 10.1177/0146621619858674.

[61] L. Xu, C. Jiang, N. He, Y. Qian, Y. Ren, and J. Li, "Check in or not? A stochastic game for privacy preserving in point-of-interest recommendation system," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 4178–4190, Oct. 2018, doi: 10.1109/JIOT.2018.2847302.

[62] D. Liu and C. Yang, "A deep reinforcement learning approach to proactive content pushing and recommendation for mobile users," *IEEE Access*, vol. 7, pp. 83120–83136, 2019, doi: 10.1109/ACCESS.2019.2925019.

[63] P. Basile, C. Greco, A. Suglia, and G. Semeraro, "Deep learning and hierarchical reinforcement learning for modeling a conversational recommender system," *Intelligenza Artificiale*, vol. 12, no. 2, pp. 125–141, Jan. 2019, doi: 10.3233/IA-170031.

[64] A. Tripathi, T. S. Ashwin, and R. M. R. Guddeti, "EmoWare: A context-aware framework for personalized video recommendation using affective video sequences," *IEEE Access*, vol. 7, pp. 51185–51200, 2019, doi: 10.1109/ACCESS.2019.2911235.

[65] J. Hribar, A. Marinescu, G. A. Ropokis, and L. A. DaSilva, "Using deep Q-learning to prolong the lifetime of correlated Internet of Things devices," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2019, pp. 1–6, doi: 10.1109/ICCW.2019.8756759.

[66] T. T. Nguyen, "A multi-objective deep reinforcement learning framework," *Eng. Appl. Artif. Intell.*, vol. 96, pp. 1–17, Nov. 2018, doi: 10.1016/j.engappai.2020.103915.

[67] J. Lee, B. Oh, J. Yang, and U. Park, "RLCF: A collaborative filtering approach based on reinforcement learning with sequential ratings," *Intell. Autom. Soft Comput.*, vol. 8587, pp. 1–6, Oct. 2016, doi: 10.1080/10798587.2016.1231510.

[68] J. Zhao, H. Li, L. Qu, Q. Zhang, Q. Sun, H. Huo, and M. Gong, "DCFGAN: An adversarial deep reinforcement learning framework with improved negative sampling for session-based recommender systems," *Inf. Sci.*, vol. 596, pp. 222–235, Jun. 2022, doi: 10.1016/j.ins.2022.02.045.

[69] L. Huang, M. Fu, F. Li, H. Qu, Y. Liu, and W. Chen, "A deep reinforcement learning based long-term recommender system," *Knowl.-Based Syst.*, vol. 213, Feb. 2021, Art. no. 106706, doi: 10.1016/j.knosys.2020.106706.

[70] Y. Lin, F. Lin, W. Zeng, J. Xiahou, L. Li, P. Wu, Y. Liu, and C. Miao, "Hierarchical reinforcement learning with dynamic recurrent mechanism for course recommendation," *Knowl.-Based Syst.*, vol. 244, May 2022, Art. no. 108546, doi: 10.1016/j.knosys.2022.108546.

[71] Z. Chai, Y.-L. Li, Y.-M. Han, and S.-F. Zhu, "Recommendation system based on singular value decomposition and multi-objective immune optimization," *IEEE Access*, vol. 7, pp. 6060–6071, 2019, doi: 10.1109/ACCESS.2018.2842257.

[72] O. L. de Weck, "Multiobjective optimization: History and promise," in *Proc. 3rd China-Japan-Korea Joint Symp. Optim. Struct. Mech. Syst.*, 2004, p. 34, doi: 10.1109/TEVC.2009.2017515.

[73] M. Hausknecht and P. Stone, "Deep recurrent Q-learning for partially observable MDPs," in *Proc. AAAI Fall Symp. Ser.*, 2015, pp. 29–37.

[74] Z. Jia, Q. Gao, and X. Peng, "LSTM-DDPG for trading with variable positions," *Sensors*, vol. 21, no. 19, pp. 1–12, 2021, doi: 10.3390/s21196571.

[75] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," 2016, pp. 1–10, *arXiv:1511.06939*.

[76] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[77] Y. K. Ee, N. M. Sharef, R. Yaakob, and K. A. Kasmiran, "LSTM based recurrent enhancement of DQN for stock trading," in *Proc. IEEE Conf. Big Data Analytics (ICBDA)*, Nov. 2020, pp. 38–44, doi: 10.1109/ICBDA50157.2020.9289832.

[78] F. M. Harper and J. A. Konstan, "The MovieLens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, pp. 1–19, Jan. 2016.

EE YEO KEAT received the B.S. degree in science instrumentation from the Department of Physics, Faculty of Science, Universiti Putra Malaysia, where he is currently pursuing the M.Sc. degree. His research interests include reinforcement learning, recommendation systems, and multiobjective optimization.

NURFADHLINA MOHD SHAREF is an Associate Professor with the Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Malaysia. Her research interests include text mining, recommendation systems, and data science. Her current projects are multiobjective deep reinforcement learning, multitask deep learning for multiclass tweets classification, and multicriteria recommendation system.

RAZALI YAAKOB (Member, IEEE) received the bachelor's degree in computer science and the master's degree in computer science from Universiti Putra Malaysia, in 1996 and 1999, respectively, and the Ph.D. degree from the University of Nottingham, U.K., in 2008. Currently, he is the Dean with the Faculty of Computer Science and Information Technology, Universiti Putra Malaysia. His research interests include artificial neural networks, pattern recognition, and evolutionary computation in game playing. He is a member of the Intelligent Computing Group at the faculty.

KHAIRUL AZHAR KASMIRAN received the Ph.D. degree from the University of Sydney, Australia, in 2012. He is a Senior Lecturer with the Department of Computer Science, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Malaysia. His interests include deep learning, reinforcement learning, performance engineering, formal verification, and software development.

ERZAM MARLISAH is a Senior Lecturer with the Department of Computer Science, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Malaysia. His research interests include both theoretical and applied artificial intelligence, machine learning, reinforcement learning, and evolutionary computing.

NORWATI MUSTAPHA received the B.Sc. degree in computer science from Universiti Putra Malaysia, in 1991, the M.Sc. degree in information systems from the University of Leeds, U.K., in 1995, the Ph.D. degree in artificial intelligence from University Putra Malaysia, in 2005. She is an Active Researcher in the area of data mining, web mining, social networks and intelligent computing.

MASLINA ZOLKEPLI received the bachelor's and master's degrees in computer science from Universiti Putra Malaysia, in 2007 and 2010, respectively, and the Ph.D. degree in computational intelligence and systems science from Tokyo Institute of Technology, Japan, in 2015. She is currently a Senior Lecturer with the Department of Computer Science, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia. Her research interests include business analytics, fuzzy systems, and computational intelligence.