Proposed Recommender System For Solving Cold Start Issue Using k-means Clustering and Reinforcement Learning Agent

Ahmed bahaaulddin A.alwahhab
Informatics Dept.
Technical College of management
Middle Technical University

Baghdad, Iraq
ahmed80.ab@gmail.com

2020 2nd Annual International Conference on Information and Sciences (AiCIS) | 978-1-7281-9169-0/20/$31.00 ©2020 IEEE | DOI: 10.1109/AiCIS51645.2020.00013

Abstract—**The cold start problem for new users is making a real challenge in the recommender system's operation to provide suggestions for a new user. This paper suggests a reinforcement learning recommender system that provides the automatic systems of multi-armed bandits which are learning and improving its efficiency from experience without being explicitly programmed. So, a movie recommender system is built using the K-Means Clustering to cluster the dataset and epsilon greedy reinforcement learning agent to manage multi-armed bandits recommendation process. The recommender system consists of multi-armed bandits that connect to five clustered datasets represents five movie genres and that was the first contribution. The second contribution is checking whether the NDCG is sufficient in measuring the quality of services in multi-armed bandits' recommender systems. The proposed recommender system has been tested using a movie lens 100-K dataset. The system measured using accumulative gain, RMSE, and NDCG. The results are showing efficiency to learn new user preferences**.

Keywords: Recommender system, clustering, bandit, reinforcement learning, epsilon greedy.

I. INTRODUCTION

The Electrocardiogram (ECG) is just a depiction of the electrical vitality of the heart, ordinarily sketched on a sheet for simpler observations. The muscle of the heart is constricted in restraint to de-polarization of its cells. It is the aggregate of the electrical vitality, when magnified and registered for a small number of seconds which we define as an ECG [1-3]. Digital information, derived from biomedical signals, is massively increased, in particular (ECG), which used for observation and diagnostic tasks. Thus, the saving of computerized data is substantial. The saving process has restrictions that made the compression of ECG is a significant subject of study in the signal processing field. Besides to aforementioned purposes, there are several benefits of signals compression, for instance, the sending time of the real-time signal is reduced, and the transmission will be inexpensive. There are particular numbers of a specimen of the signal that are abundant and can be eliminated with keeping the significant characteristics of the signal. The main aim of ECG compression approaches is to attain a minimized data rate and to maintain the necessary medical information in the recovered signal [4]. This study will concentrate on ECG Compression Schemes based on various transforms.

II. RELATED WORKS

Li [3] designs depend on personalized contextual bandit recommendations for news articles. The proposed LinUCB stands for linear upper confidence bound, which is an algorithm that represents an extension of the UCB algorithm. It selects the news based on mean and standard deviation. It also depends on a factor α to balance between the exploration/exploitation actions. Despite the similarity between the proposed system in using bandit with clustered data, there are the only differences that Li worked on the news dataset. In contrast, this proposed system worked on the movie-Lense dataset [3]. Caron and Bhagat [5] exploit social information parts into bandit algorithms to tackle the cold-start problem using an additional information source[5]. Lacerda et al. [2015] address users as bandit's arms to provide recommendations for daily-deals. Their paper considers strategies for splitting users into exploration and exploitation parts [6]. Cricia Z. et al. [7] proposed an algorithm depends on two phases (i) computing and updating prediction models phase (ii) recommendations phase. The recommendation process depends on the bandit to choose the prediction model and determine the items to recommend to a user. The algorithm uses a set of item possibilities the use a multi-armed bandit to select an item for a user at each time[7]. Wang et al, [8]. The used matching system between user preferences and latent vector of items that have been chosen before by the user. The update process of each bandit's weight is done using particle strategy by updating each arm's weight. Despite that, the movie lens dataset was used to test the algorithm; unfortunately, the CTR score was used to evaluate the algorithm[8].

In this research, the five bandits have been created and modeled on clustered datasets under five genres of movies from the movie lens dataset of 100-k ratings.

III. THE MULTI-ARMED BANDIT PROBLEM

The multi-armed bandits' issue is a classic reinforcement learning issue where are given n of machines (bandits) slots. Each bandit has an arm to be pulled by a player/user.

978-1-7281-9169-0/20/$31.00 ©2020 IEEE
DOI 10.1109/AiCIS51645.2020.00013

Each machine has its probability distribution of success. When Pulling any bandit's arm, this will give the user a stochastic reward of either r=1 or r=0 in the failure case. Any player/user's objective is to pull the arms (bandits) one-by-one in a sequence that can maximize the total reward for that player in a long time[9].

Bandit problem algorithms are approaches to real-time/online decision making that can balance between sufficiently exploring the variant space and exploiting the optimal action. There must be a policy to balance exploration/ exploitation. This policy begins with exploring all the bandits to define the best bandit in initial. Accordingly, the player exploits the optimal action maximizing the total reward available from that set of bandits. Simultaneously, the player can explore the other feasible bandits to gain better returns in future rounds. This issue of exploration/exploitation can be solved using the epsilon-Greedy approach or policy [9].

IV. Epsilon-Greedy approach

The epsilon greedy algorithm s the easiest type of the bandit problem-solving approach. This approach tries to compromise between two goals of exploration and exploitation using a straightforward mechanism[10]. The idea behind the epsilon greedy algorithm is that simple: if a coin return mostly tails and a person flip a coin and comes up with heads, this person should explore for a moment, but if a coin comes up with tails, this person should exploit. After an initial period of exploration (for example, 1000 trials), the algorithm greedily exploits the best option *k*, *e* percent of the time. For example, if we set *e*=0.05, the algorithm will exploit the best choice 95% of the trials while exploring the random alternatives 5% of the time. One potential solution could be to now, and a player can then explore new bandits. According to that, the system ensures that bandit system does not miss out on the arm's better choice [11]. Depending on the reinforcement learning, the state-action function Q(s, a) is defined to represent the expected future gained reward. This gain will come from state S. In the field of the multi-armed bandit; each action makes the agent closer to the terminal state. So, long term rewards are absolutely the accumulated immediate rewards and the definition of action value as:

$$Qk(a) = \frac{1}{k}\ (r1 + r2 + \cdots . + rk) \quad \text{--------- (1)}$$

Where k is the count for how many the specific bandit k bandit were chosen in the past and r represents the stochastic reward for each time that bandit was selected. The equation can be represented Q (a) recursively using the following equation:

$$Qk + 1 = Qk(a) + \frac{1}{k}(rk + 1 - Qk(a)) \text{-------- (2)}$$

So epsilon-greedy agent has to choose the bandit greedily with the maximum number of history of rewards Q(a) while selecting the same of another bandit just for exploration, so the greedy choosing depends on the equation (3):

$$a\ greedy = argmax\ a\ Qk(a) \text{---------- (3)}$$

With epsilon probability, the agent will choose an action randomly for (exploration part) in epsilon percentage while the same agent can choose an action with highest Qt (a) with probability 1-epsilon as (exploitation part). So *With probability 1- epsilon – that agent will choose action with the highest value (argmax Qt (a))* while *with probability epsilon – the agent choose an action from a set of all actions at random A. see example in figure (1)*

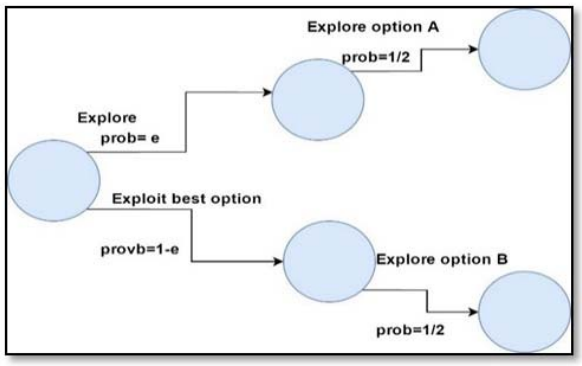

Fig. 1 *Epsilon-Greedy approach*

This policy is much better than the just greedy approach because the algorithm has an element of exploration here. In general, if two actions have very little difference between their Q values, then this algorithm will select the action which has a probability HIGHER THAN THE OTHERS[12].

V. Preprocess the dataset

Before design the multi-armed bandit recommender system, the proposed paradigm needs to implement EDA Exploratory Data Analysis before preprocessing the data (clustering the dataset) and design the data for each bandit. The EDA has to answer the crucial questions: Q1: What are the common genre of movies? , Q2: how much rating people give mostly?

The most useful question in this system is the first one. It will help arrange the bandit and organize bandits' arms between exploitation (for the most common genre) and exploration for less watched genres. The most preferred genre is drama, then 4 genres (comedy, thriller, action, and romance) as in figure 2:

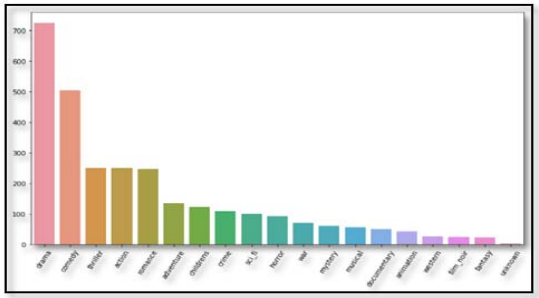

Fig. 2 list of preferred movies genres

700 users watched the drama. So depending on that analysis, the drama will be fed the first bandit with movies for exploitation. And rest 4 genres will provide the other bandits with movies for exploration with 0.05 of epsilon

amount. Answering the second question is that most people give 3 for accepted films and 4, 5 starts for most preferred movies, so the most given reward given to the bandit when accepting a film by a user will be 3 otherwise 0 for not accepted one. See figure (3)

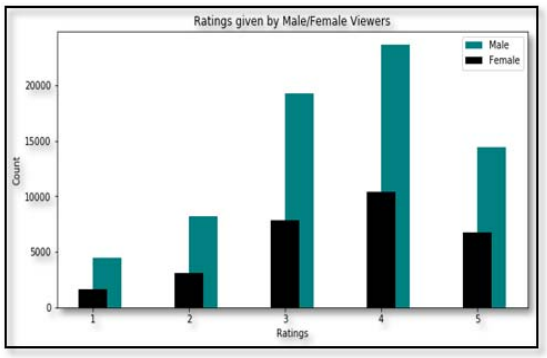

Fig. 3 Rating given by male and female users

Now, preprocess the dataset for clustering. A cluster refers to a collection of data points aggregated together because of certain similarities. K-means identifies k-means of centroids and allocates every point to the nearest cluster as small as possible. Before implementing the clustering, which is the preprocessing paradigm in the proposed recommender system. There is a method to check the best number of clusters that have to be proposed before running the k-means clustering algorithm. This checking method is known as the elbow- method. The Elbow method is a heuristic method that is often used to determine the appropriate number of clusters in a data set, which is will be clustered.

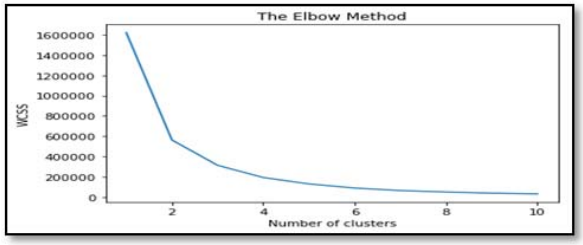

Fig. 4 elbow graph for romance movies

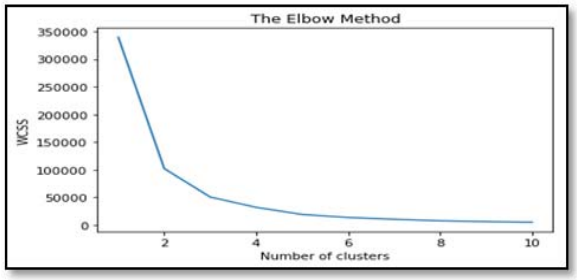

Fig. 5 elbow graph for drama movies data

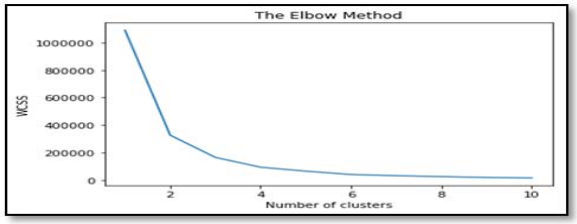

Fig. 6 elbow graph for comedy movies data

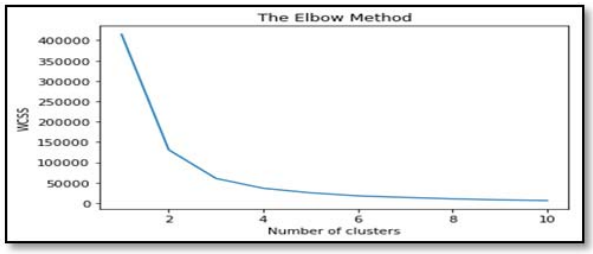

Fig. 7 elbow graph for thriller movies data

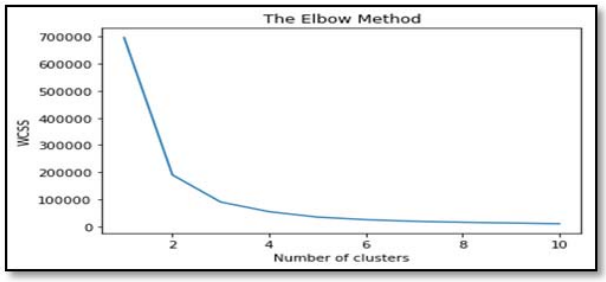

Fig. 8 elbow graph for action movies

Now implement the clustering users into four clusters according to elbow method analysis inside each of 5 genres of movies. Look at the flowchart in figure (9):

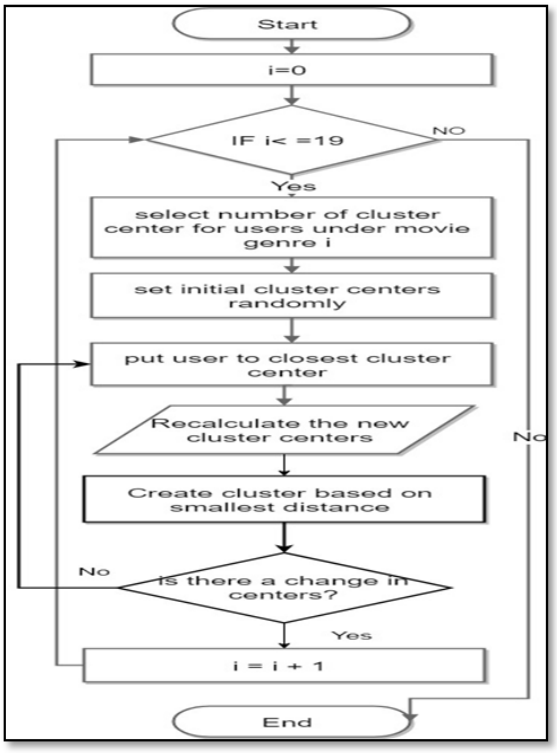

Fig. 9 k-means clustering flowchart

VI. Proposed System

The proposed system tackled the problem of the new user cold start in recommender systems. The proposed recommender system depends on improving the epsilon-greedy multi-armed bandit's recommender system. In this recommender system, the system will choose a suitable bandit according to its history of accumulative rewards. Each bandit in the proposed system represented the movie genre, so the proposed system will have 5 genres, one bandit for each of the movie genres. See figure (10) of the general diagram of the proposed recommender system

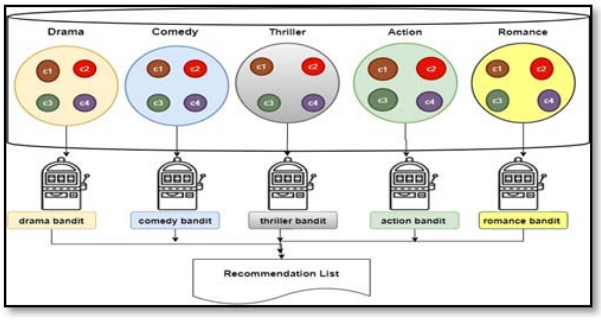

Fig. 10 general view of the multi-armed RS

The proposed recommender system depends on playing five of bandits. Each one represented a genre type and each one connected to its dataset source. Each of the datasets is clustered into 4 clusters according to the degree of preferences from users inside that genre. When a user/visitor enters the system, the recommender engine will check the preferred movie type preferred for that user. If the user/visitor likes drama movies for example, the recommender engine will make the drama bandit the better bandit for the exploitation part (greedy playing) and specifies the other bandits (other movie types) for exploration.

$$\max(Q) = 1 - e \quad \text{---------- (4)}$$

$$any\ action\ (a) = e \text{ -------- (5)}$$

e or epsilon her in equations represent the percent of exploration that has been specified to the recommender system to choose between the most prefer bandit (expiation) or choosing other bandits in random as exploration policy.

The epsilon-greedy strategy provides the chance for other bandits or the other types of movies to be recommended in a specific present. This strategy makes the recommender engine response to the variation of user mood in each of his visits. If the user 2 in one session prefers drama the engine will focus on drama bandit as user 2 gives most of the reward to that bandit (the type of movies). If the same user prefers comedy movies in another session (visit), the recommender system engine will focus on comedy bandit as exploitation policy while giving a chance for other types (bandits) in epsilon percent. The total reward will be used to measure the efficiency and user satisfaction of the system. The accumulated reward will help the recommender engine to choose the next bandit arm for the user. The total reward is presented in the following equation:

$$Q(A) = Q(A) + \frac{1}{N(A)}\,[R - Q(A)] \text{ ------- (6)}$$

The general epsilon greedy algorithm can be illustrated by the following algorithm:

ALGORITHM : EPSILON GREEDY

INPUT : NUMBER OF BANDITS

Initialize :

Q(a) =0

N(a) = 0

OUTPUT: Reward of each bandit

1- Start

2- Repeat forever:

A ←

$$\begin{cases} argmaxa\ Q(a)\ with\ probability\ e-1 \\ a\ random\ action\ with\ probabiity\ e \end{cases}$$

R← bandit(A)

N(A)← N(A)+1

$$Q(A) = Q(A) + \frac{1}{N(A)}\ [R - Q(A)]$$

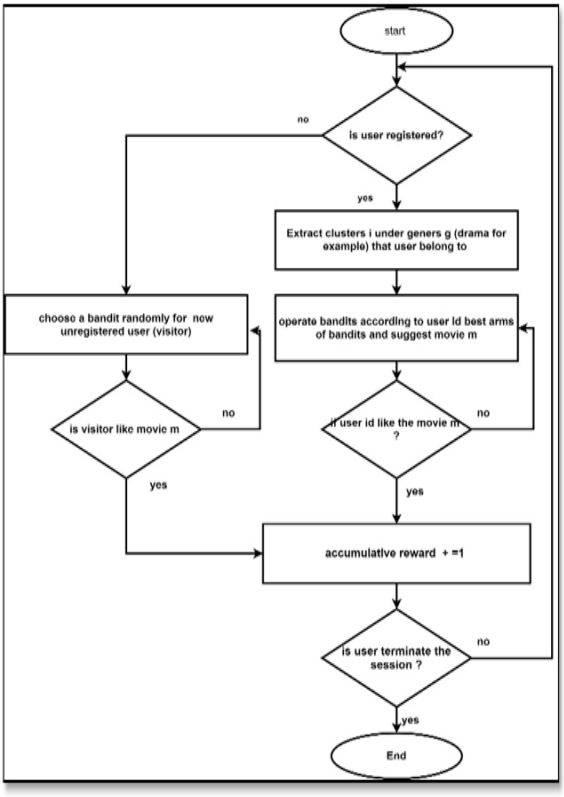

Fig. 11 General flowchart of the proposed multi-armed recommender system

The bandit chooses a movie from the dataset that the guest or user i belong to the cluster. The proposed and it using the proposed system improves the design of Epsilon-greedy k-means clustering on the dataset, by recommender system bandit is shown in figure (11)

So, dividing and organizing the data across bandits will increase the performance and efficiency of the multi-armed bandits by making the system understand the user/guest efficiency and rapidly.

VII. Results

There are three measurements to measure the performance of the proposed multi-bandit recommender system; first, the accumulative reward for each user. Second, the root means squared error (RMSE), and third the (Normalized discounted cumulative gain) NDCG.

To measure the accumulative gain, a sample of four users has been taken; these are user 10, user 20, user 77, user 1, and user 600. See figure 12 below

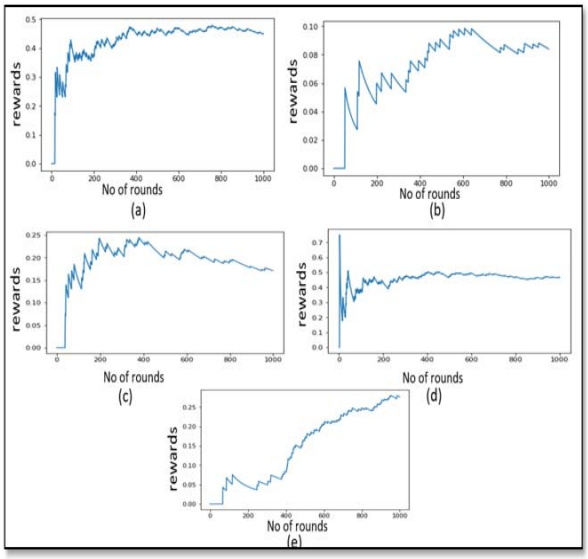

Fig. 12 accumulative gain for users (a) 10, (b) 20, (c) 77, (d) 1, (e) 600

The cumulative gain is different from one user to another because it represents the total gain from playing the five bandits representing the (drama, comedy, action, thriller, and romance). The gain of some users between 0 and 1. While for user number 77 (figure 12 c) it's between 0 and 0.25 which is represented in the x-axis of the figure while the y-axis represents the number of playing round for all bandits that are 1000. This variation belongs to the varying cumulative reward for each user's arm caused by the number and type of movies that each user watched and liked. The accumulative rewards are increased from the beginning as the epsilon amount is 0.05 so just 5% will be for explore new arm. At the same time, most of the play will depend on bandit that contains user preferences or ensure the reward from a specific user. Now let see the results of RMSE for each user in the table (1)

TABLE 1 RMSE for user's sample

User_id	RMSE
10	0.74
20	0.70
77	0.77
1	0.74
600	0.71

The system tested for 200 users as test set users, and the average of RMSE for these users was 0.74. The RMSE for the movie-Lense dataset roughly is common.

The third measurement is to measure the quality of service of the recommender system with NDCG. Also, NDCG is calculated for each of the users (10, 20, 77, 1, and 600) as below in figure (13):

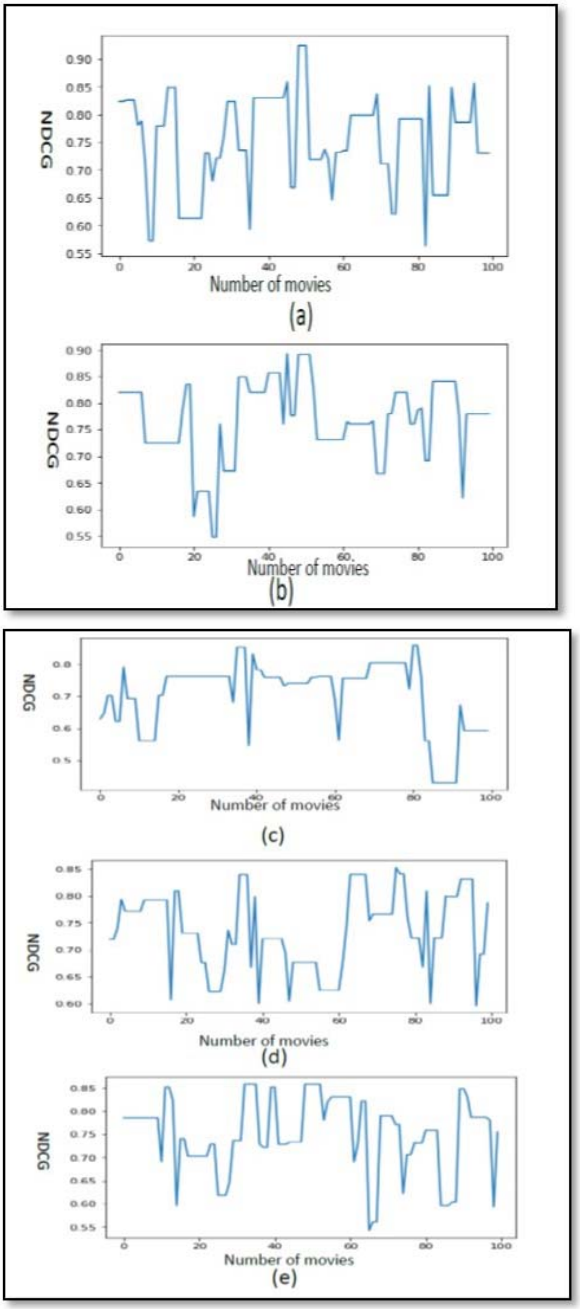

Fig. 13 NDCG for users (a) 10, (b) 20, (c) 77, (d) 1, (e) 600

As noticed from NDCG figure 13 (a, b, c, d, e), the NDCG for all users is wobbling because the recommender system

depends on exploration/exploitation by choosing movies randomly from each bandit dataset separately, and the system agent will be learned progressively from the user as the iteration increased. NDCG is increased in moments because a user may not reward a suggested movie. Now the second test is testing the traditional system of multi-armed bandit that users just two bandits. First bandit work with data that contains movies that are rated between 3 and 5 stars. The second bandit works with data that has movies rated with less than 3 stars by audiences. The first bandit represents exploitation as it works with exact successful movies. In contrast, the second bandit represents exploration because that bandit takes its movies that have not been accepted by all the users. So, this set will be for the exploration of new and strange choices. Let test for the same four users that have been tested for the proposed system with the same users in figure 14:

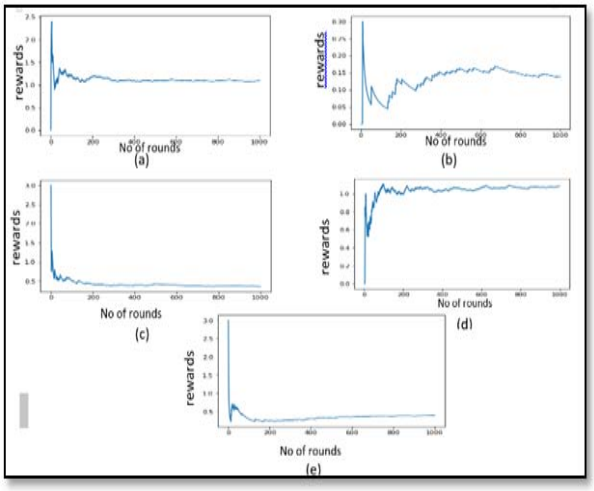

Fig. 14 accumulative gain for users (a) 10, (b) 20, (c) 77, (d) 1, (e) 600 using traditional method

The accumulative reward is increased rapidly then decreased because the bandit system has two choices or two bandits. The first choice is the best movies and the second one are not well- known movies (the second bandit). It's clear from the comparison between accumulative reward with the proposed recommender system and the traditional bandit system that the accumulative reward increased gradually in figure (12) in figures (a,b,c,d,e) while it's decreased after while in the traditional bandit recommender system in figures (14 within a, b, c , d, e. each figure represent user test) . As a result, the proposed system is better than the classic multi-armed bandit recommender system. The proposed system depends on EDA to arrange the data for bandit's response better for users than the previous way.

Now the NDCG for the traditional bandit system can be noticed not decreased as in the previous one figure (13). The NDCG scores for each user keep high in most of the trials. See figure 15 (for user's figures a, b, c, d, and e) below:

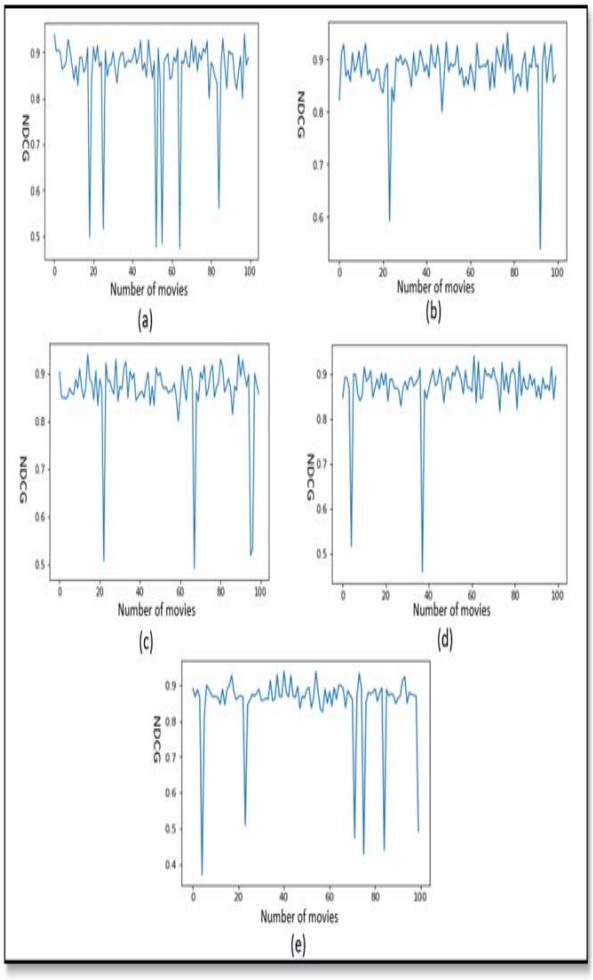

Fig 15 NDCG for users (a) 10, (b) 20, (c) 77, (d) 1, (e) 600 under traditional algorithm

Figure (15 a) for user 10 the NDCG not decreased in trials between 30 and 50, for example, while its amount decreased rapidly for the same user in the proposed algorithm. The same thing NDCG did not decrease for user 20 in figure (15 b) between trials 30 and 80 NDCG not reduced it is still in the same range between 0.8 and 0.9. The same thing was noticed for users 1 figure (15 D), Figure (15 c) 77, and 600 in figure (15 e). NDCG has not decreased in most of the trials. The RMSE for the traditional multi-armed bandit recommender system was about 0.86 which is higher than the RMSE in the proposed algorithm. The NDCG measurement may not be suitable for recommender systems that depend on reinforcement learning principles because the user's reward depends on the randomly chosen movie provided for the user. Figures below show the reader a comparison between the proposed algorithm and the traditional one must be discussed to have a clearer view. Figure (16) represents a comparison in RMSE between the proposed multi-armed bandit and traditional multi-armed algorithm as follows

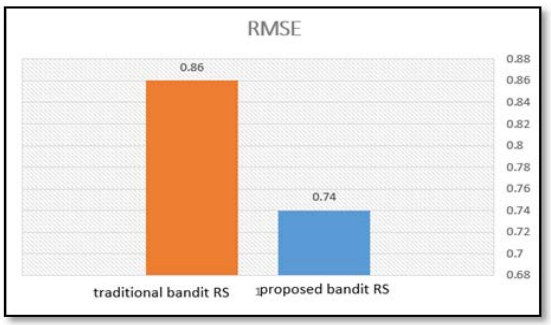

Fig. 16 RMSE comparison between proposed and traditional algorithms

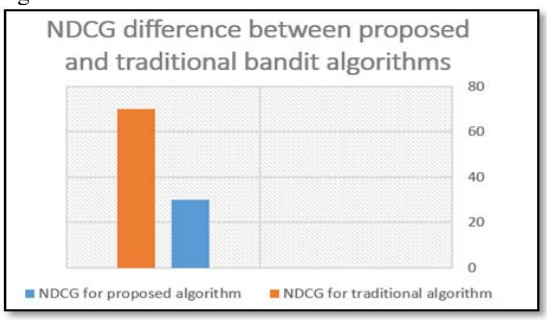

Fig. 17 NDCG difference between proposed and traditional algorithms

NDCG caused by proposed system test was less about 30% from the traditional algorithm. In spite of that the NDCG appear not good metric to be used in multi-armed bandit recommender system in general because the RS fluctuate between exploration/exploitation.

Accumulative gain in figure (18) below is clearly higher in the proposed RS than traditional one because the scope of search for movies within five movie's genres in bandits is more accurate than its counterpart with just only two bandits.

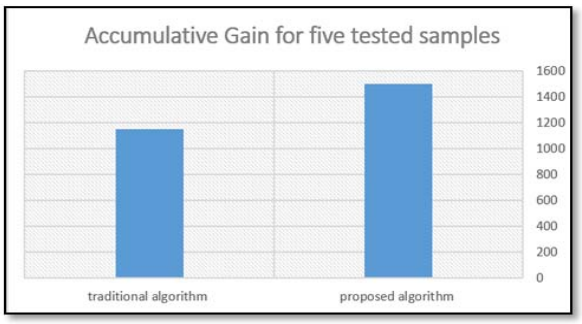

Fig. 18 Accumulative Reward comparison between proposed and traditional algorithms

VIII. CONCLUSION

The reinforcement learning multi-armed bandit can be used to learn the preferences of a user gradually. This paradigm will lead to overcome the cold start problem in the recommender system for new users. This research paper wants to answer two questions. First, whether organizing the data and clustering the users under genres of most-watched/used movies or items helps to increase the efficiency of multi-armed bandits? According to the results, and depending on the comparison between multi-armed bandit systems both the proposed recommender system that is working with a clustered dataset and the second without clustered data (traditional) multi-armed bandit. It's clear that using clustering data according to the EDA process using k-means clustering to 5 genres and 4 clusters under each genre in the movie-lens dataset causes the accumulative gain to be increased. That's mean the reward from users increased through the trials of playing the bandits by the user. While the RMSE is a decreased proposed system. The second question is whether NDCG is suitable for measuring the quality of recommender systems that depend on reinforcement learning (multi-armed bandits)? To measure the quality of recommender system service NDCG is used. But according to the tests of this paper NDCG is not suitable for a recommender system that relies on reinforcement learning agents (bandits) because it serving choices randomly. Despite that NDCG was decreased during the test for the first 100 recommendations. The high ramps in NDCG coming from some chosen movies randomly from the bandit agent were not rewarded by the user in a specific session.

REFERENCES

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use “Ref. [3]” or “reference [3]” except at the beginning of a sentence: “Reference [3] was the first ...”

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors’ names; do not use “et al.”. Papers that have not been published, even if they have been submitted for publication, should be cited as “unpublished” [4]. Papers that have been accepted for publication should be cited as “in press” [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

[1] J. Mary *et al.*, "Bandits and Recommender Systems To cite this version : HAL Id : Hal-01256033," 2016.

[2] M. Elahi, "Cold Start Solutions for Recommendation Systems Music recommender systems View project EXTRA: EXpertise-Boosted Model for Trust-Based Recommendation System Based on Supervised Random Walk View project," no. April, 2019, doi: 10.13140/RG.2.2.27407.02725.

[3] L. Li, D. D. Wang, S. Z. Zhu, and T. Li, "Personalized news recommendation: A review and an experimental investigation," *J. Comput. Sci. Technol.*, vol. 26, no. 5, pp. 754–766, 2011, doi: 10.1007/s11390-011-0175-2.

[4] G. Koutrika, "Recent advances in recommender systems: Matrices, bandits, and blenders," *Adv. Database Technol. - EDBT*, vol. 2018-March, pp. 517–519, 2018, doi: 10.5441/002/edbt.2018.61.

[5] S. Caron and S. Bhagat, "Mixing bandits: A recipe for improved cold-start recommendations in a social network," *Proc. 7th Work. Soc. Netw. Min. Anal. SNA-KDD 2013*, 2013, doi: 10.1145/2501025.2501029.

[6] A. Lacerda, "Multi-Objective Ranked Bandits for Recommender Systems," *Neurocomputing*, vol. 246, pp. 12–24, 2017, doi: 10.1016/j.neucom.2016.12.076.

[7] C. Z. Felício, K. V. R. Paixão, C. A. Z. Barcelos, and P. Preux, “A multi-Armed bandit model selection for cold-start user recommendation,” *UMAP 2017 - Proc. 25th Conf. User Model. Adapt. Pers.*, no. Figure 1, pp. 32–40, 2017, doi: 10.1145/3079628.3079681.

[8] Q. Wang *et al.*, "Online Interactive Collaborative Filtering Using Multi-Armed Bandit with Dependent Arms," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 8, pp. 1569–1580, 2019, doi: 10.1109/TKDE.2018.2866041.

[9] J. M. White ."***Bandit algorithms for web site optimization***", Oreilly publications. 2013.

[10] F. Guillou *et al.*, "Collaborative Filtering as a Multi-Armed Bandit To cite this version : HAL Id : hal-01256254 Collaborative Filtering as a Multi-Armed Bandit," 2016.

[11] D. Bouneffouf, S. Parthasarathy, H. Samulowitz, and M. Wistuba, "Optimal exploitation of clustering and history information in multi-armed bandit," *IJCAI Int. Jt. Conf. Artif. Intell.*, vol. 2019-August, pp. 2016–2022, 2019, doi: 10.24963/ijcai.2019/279.

[12] M. Tokic, "Adaptive ε-greedy exploration in reinforcement learning based on value differences," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6359 LNAI, no. April, pp. 203–210, 2010, doi: 10.1007/978-3-642-16111-7_23.