Received: October 14, 2022. Revised: November 18, 2022.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

User Feature Similarity Supported Collaborative Filtering for Page Recommendation Using Hybrid Shuffled Frog Leaping Algorithm

Duraisamy Deenadayalan[1]* **Athiyappagounder Kangaiammal[2]**

[1]*PG and Research Department of Computer Science, Government Arts College (Autonomous), Salem, Tamilnadu, India*
[2]*Department of Computer Applications, Government Arts College (Autonomous), Salem, Tamilnadu, India*
* Corresponding author's Email: deena.duraisamy@gmail.com

Abstract: The rapid growth of e-commerce has caused product overload where customers on the Web are no longer able to effectively choose the products they are exposed to. To overcome the product overload of online shoppers, a variety of recommendation methods have been developed. Collaborative Filtering (CF) is the most successful recommendation method, but its widespread use has exposed some well-known limitations, such as sparsity and scalability, which can lead to poor recommendations. This work proposes a recommendation methodology based on Web Usage Mining (WUM), and machine learning methods to enhance the recommendation quality and the system performance of current CF-based recommender systems. WUM populates the rating database by tracking learners' behaviors on the Web, thereby leading to better quality recommendations. The data is collected from user profile and their preferences and also the weblink and usage. Based on the CF and WUM data, the recommendations are provided. The Random Forest (RF) and Extreme Gradient Boosting (XGBoost) classifiers is used to improve the performance of searching for nearest neighbors through Shuffled Frog Leaping Algorithm (SFLA). Experimental results show that the proposed model is effective and can enhance the performance of recommendation. Results show that the proposed SFLA-XGBoost has higher average sensitivity: rating <=15 for CF by 11.49% for RF, by 7.52% for XGBoost and by 4.58% for proposed SFLA-RF respectively.

Keywords: E-learning, Recommender system, Collaborative filtering (CF), Web mining usage (WUM), Random forest (RF), Extreme gradient boosting (XGBoost), Shuffled frog leaping algorithm (SFLA).

1. Introduction

Nowadays, more and more people have their own smart phone, tablet PC and other intelligent terminals. This has enabled them to spend more time in accessing all kinds of social networks (such as Facebook and Twitter) and e-commerce sites (such as Amazon and eBay). However, the huge amount of available information and products makes them overwhelmed and indecisive. Users have to spend more time and energy in searching for their expected information [1]. Even then, they cannot get satisfactory results. Fortunately, the behaviours of users can be tracked and recorded on the social networks and e-commerce sites. This makes it easier to analyze the preference of users. In this regard, recommender systems are used to recommend information of user expectations and provide personalized services through analyzing the user behaviors, such as the recommendation of photo groups in Flickr, the books in Amazon, videos in YouTube, and results in the Web search.

Currently, popular recommendation algorithms are mainly divided into content-based recommendation, Collaborative Filtering (CF) recommendation, hybrid recommendation, and other algorithms [2]. Content-based recommendation utilizes a series of discrete features of items, e.g., the genres, directors, and actors in movies, to generate recommendation. CF recommendation aims to calculate a list of interesting items to target users based on the preferences of their like-minded neighbourhood [28]. These two approaches are often

combined to make hybrid recommendation. As is known to us all, CF is one of the most widely used and successful technologies in personalized recommendation systems. The core idea is that the past preference behaviours of users have a significant influence on their future behaviours, and their previous behaviours are basically consistent with future behaviours. Generally speaking, the similarity between users is estimated according to the user's historical behaviour. And then, according to the evaluation of the neighbours with high similarity to the target users, the target users are predicted whether to be interested in the item.

With the rise of artificial intelligence in recent years, the government and all occupations have vigorously developed artificial intelligence to promote the development of society and technological progress [3]. Relying on the advantage that the Internet can transcend time and space, the traditional teaching model has gradually changed, the way of learning knowledge is also richer and multilevel, and education has become simpler and more effective. The network of online education began to flourish and become popular. The development of online education and the level of technological development are closely related to the change of educational philosophy, the upgrade of people's demand for user education, and the change of lifestyle. Group lets' research group proposed the key technology of the recommendation system: collaborative filtering, which is the key technology of online education. Since then, personalized recommendations have begun to flourish. A good recommendation system gives users a sense of belonging, makes them trust the system, and provides them with good personalized service.

Learning Management Systems (LMSs) [4] are web based tool for delivering, monitoring and managing the educational courses or training programs. LMSs facilitate the managing and online collaboration of educational records. Some of the popular open sources LMSs are Moodle, Sakai and Atutor. Tutors often use LMSs to administer their courses, which include communication with their students via forums, chats and massage services available within LMSs. LMSs also offers a framework which can promote information exchange and knowledge sharing between the students' registered in a course, allow tutor and student communication, generate learning material, start discursions and promote cooperative learning using the social media features available in LMSs, such as forums, chats, file storage databases, message and news sharing services and others. LMSs stores log information about students' activities, that include access logs, reading, writing, participating in assessment related activities such as taking tests, uploading the assignments/case studies, performing different group work, including interacting with peers. LMSs' also provides a database that stores personal information about the student (profile), academic results, students' interaction data, etc. However, due to the so many of these daily activities, LMSs generated large amount of data, and it is a tedious task to manage this manually, so tutors will prefer to have some tools which can help them in specific tasks, preferably on a regular basis.

Several researchers working in the field of LMS to enhance students learning experiences highlighted the need of RSs' for LMS [5], so as to address the following challenges in LMSs':

- ✓ Difficulty in sharing the learning resources;
- ✓ High redundancy of learning material;
- ✓ Personalization of information;
- ✓ Information overload which is the ever increasing volume of digital information particularly on the web, and due to this reason it has becomes extremely more and more difficult for learners to find suitable items to satisfy a particular need;
- ✓ Learning isolation.

Web Usage Mining (WUM) [6] is the process of extracting knowledge from Web user's access data by exploiting data mining technologies. It can be used for different purposes such as personalization, recommendation system improvement and site etc. Since the web data is semi structured or structured a semantic knowledge of the data will be helpful in understanding the data. Semantic means that the meaning of data can be discovered by computers. Currently, many machine learning methods have been used for the model-based CF, such as the Back ward Propagation (BP) neural network, Adaptive learning, and linear classifier [7]. Currently, CF based recommendation techniques have been applied in a variety of areas, such as music recommendation, news recommendation, product recommendation, etc. Random Forest (RF) and XGBoost classifiers is a widely adopted machine learning method. Many heuristic algorithms have thus been used for the parameter optimization of RF and XGBoost, such as the Grid Search (GS), Genetic Algorithm (GA), and Particle Swarm Optimization (PSO). Comparing with other algorithms, SFLA is recognized to have merits of strong global search capability and ease of implementation. But the standard SFLA also has some demerits. It often pre-matures into the local optimum and has slow convergence speed. In this work, proposes the SFLA with RF and XGBoost algorithms for CF based recommender systems. The remaining part of the investigation is organized into

the following sections. Section two discusses related works in literature. Section three explains various methods used in the work. Section four discusses experimental results and section five concludes the work.

2. Related works

An important factor affecting the performance of CF for recommendation systems is the data sparsity of the rating matrix caused by insufficient rating data. Improving the recommendation model and introducing side information are two main research approaches to address the problem. Duan et al., [8] combined these two approaches and proposed the Review-Based Matrix Factorization method. The method consists of two phases. The first phase is review-based CF, where an item-topic rating matrix is constructed by the feature-level opinion mining of online review text. This rating matrix is used to derive item similarities, which can be used to infer unknown users' ratings of the items. The second phase consists of rating imputation, where it first fill some of the empty elements of the user-item rating matrix, then conduct matrix factorization to learn the latent user and item factors to generate recommendations. Experiments on two actual datasets show that the method improves the accuracy of recommendation compared with similar algorithms.

Similarity calculation is the most important basic algorithm in CF recommendation. It plays an important role in calculating the similarity between users (items), finding nearest neighbors, and predicting scores. However, the existing similarity calculation is affected by over reliance on item scores and data sparsity, resulting in low accuracy of recommendation results. Jiang et al., [9] proposed a personalized recommendation algorithm based on information entropy and Particle Swarm Optimization (PSO), which takes into account the similarity of users' score and preference characteristics. It uses random PSO to optimize their weights to obtain the comprehensive similarity value. Experimental results on public data sets show that the proposed method can effectively improve the accuracy of recommendation results on the premise of ensuring recommendation coverage.

Puraram et al., [10] proposed a hybrid method of PSO and K-Means algorithm to improve the user's dietary behavior clustering and using Principal Component Analysis (PCA) to reduce the data dimension. Moreover, the User-Based CF technique is used to predict the rating of relevant Thai food menus and recommendation. The experimental result shows the hybrid method improves the clustering performance from three models: Hierarchical Clustering, K-Means, and K-Means with PCA, in terms of silhouette coefficient score. In addition, the hybrid method improves the Davies-Bouldin index score by 44%, 19%, and 17% compared to those models, respectively. The rating prediction result shows the hybrid method outperforms the other methods.

Yue et al., [11] developed a Modified Collaborative Filtering (MCF) algorithm with improved performance for recommendation systems with application in predicting baseline data of Friedreich's Ataxia (FRDA) patients. The proposed MCF algorithm combines the individual merits of both the User-Based CF (UBCF) method and the Item-Based CF (IBCF) method, where both the positively and negatively correlated neighbors are taken into account. The weighting parameters are introduced to quantify the degrees of utilizations of the UBCF and IBCF methods in the rating prediction, and the PSO algorithm is applied to optimize the weighting parameters in order to achieve an adequate trade-off between the positively and negatively correlated neighbors in terms of predicting the rating values. To demonstrate the prediction performance of the proposed MCF algorithm, the developed MCF algorithm was employed to assist with the baseline data collection for the FRDA patients. The effectiveness of the proposed MCF algorithm was confirmed by extensive experiments and, furthermore, it is shown that the algorithm outperforms some conventional approaches.

CF is one of the primary applications that researchers use for the prediction of the accuracy rating and recommendation of objects. Various experts in the field are using methods like Nearest Neighbors (NN) based on the measures of similarity. However, similarity measures use only the co-rated item ratings while calculating the similarity between a pair of users or items. Zubair & Al Sabri [12] presented two standard methods used to measure similarities are Cosine Similarity (CS) and Person Correlation Similarity (PCS). However, both are having drawbacks, and the present piece of the investigation will approach through the optimized Genetic Algorithms (GA) to improve the forecast accuracy of RS using the merge output of CS with PCS based on GA methods. The results show GA's superiority and its ability to achieve more correct predictions than CS and PCS.

Conventional recommender systems often utilize similarity formulas to identify similarities between active users and others to predict the rating of the unseen items. Existing optimization algorithms seek to find the weights and coefficients affecting these

similarities. Houshmand Nanehkaran et al., [13] implemented in R in the GACFF package, shifts away from this view and directly uses the continuous GA to find optimal similarities in big data (e.g., Movielens 1M and Netflix datasets) to improve the performance of UBCF recommendation systems. First, by identifying the users who are the nearest neighbors along with their number, the number of genes in a chromosome is determined. Each gene represents the similarity between a neighboring user and an active user. This GA is independent of the size of the data. The method provides optimal solutions more quickly by estimating the starting points. Moreover, the genetic metric provides better results and recommendations than previous ones in terms of runtime and quality measures (i.e., mean absolute error, coverage, precision, and recall).

Houshmand‐Nanehkaran et al., [14] provided a list of the best items for recommending in less time. The Fuzzy-Genetic CF (FGCF) approach recommends items by optimizing fuzzy similarities in the Continuous GA (CGA). In this method, first, the crisp values of user ratings are converted to fuzzy ratings, and then the fuzzy similarities are calculated. Similarity values are placed into the genes of the GA, optimized, and finally, they are used in fuzzy prediction. Therefore, the fuzzy system is used twice in this process. Experimental results on RecSys, Movielens 100 K, and Movielens 1 M datasets show that FGCF improves the CF recommender system performance in terms of quality and accuracy of recommendations, time and space complexities. The FGCF method is robust against the sparsity of data due to the correct choice of neighbours and avoids the users' different rating scales problem but it not able to solve the cold-start challenge.

Du et al., [15] proposed a hybrid recommendation algorithm named K-GBDT which combined KNN and Gradient Boosting Decision Tree (GBDT). Firstly, it used KNN to obtain the similar information of the target user and item for preliminary prediction. At the same time, it mined users' basic information features and potential features of users and items from the original data. Then, it tried to adopt XGBoost, LightGBM and CatBoost algorithm which were proposed base the idea of GBDT to build regression model. Finally, the target user's rating for the movie was predicted with the trained model, and the recommendation list was generated based on the predicted results. Compared with some classic recommendation algorithms such as NMF, Slope One, Co-Clustering and etc. Based on MovieLens datasets, results show that the mean absolute error and root mean square error of this algorithm are both low, and its recommendation accuracy is higher.

As observed from the literature survey, most of the recommender systems presented are based on CF, content, or hybrid methods. Web-based recommender system using clustering methods to find recommendations. Thus, we proposed a Hybrid personalized user feature similarity supported CF for page recommendation. The advantages of both CF and WUM is combined to provide the recommendations.

3. Methodology

The data collected are from user profile and their preferences are tabulated as shown in Table 1.

The Collaborative Filtering (CF) method usually requires users to explicitly input ratings about pieces of information. These ratings are then used to compute pairwise correlation coefficients among existing uses. The correlation coefficient is the

Table 1. Learners' data and the weblink data collected

User_id									
Gender	Male	Female							
Exp	0	5	10						
Page content	Basic	Intermedia	High						
page_id									
time_on_page_second									
Page Rating	1 to 5								
Location									
Education									
Previous_course	HTML/C	Javascript	Linux	Cloud	C	C++	Java	PHP	Python
Degree									
Department									

measure of the how similar two users are. The system can make prediction or recommendation based on the correlation coefficients. The Fig. 1 shows a system architecture of personalization recommendation. The architecture mainly includes two parts: offline process and online process. The offline process includes data preparation and WUM and the online process is made up of recommendation engine. The offline process is called model acquisition phase and online process is called model application phase [16].

WUM will put a key emphasis on forecasting the customer's behavior each time they use the web. This mining's acquired data is merely the secondary kind of data that is obtained from the web as a result of the user's web interactions with the web. While this mining's data is generally of an extensive variety, it will classify the data on the basis of the data usage. The recommender systems will provide the user recommendations based on the users' past web browsing history as well as other variables.

The hybrid CF-WUM model will predict the recommendations by using both the weblink and page_id association model as well as the user preference model. For a specific user, a personalized recommendation list will be generated by scoring each candidate page_id for the user, and by picking the best match. This score must reflect the degree of similarity between the user preference as well as the weblink association. Later, the recommendations are acquired based on the classifiers.

In this section, the RF, XGBoost, SFLA, SFLA with RF and SFLA with XGBoost methods are discussed.

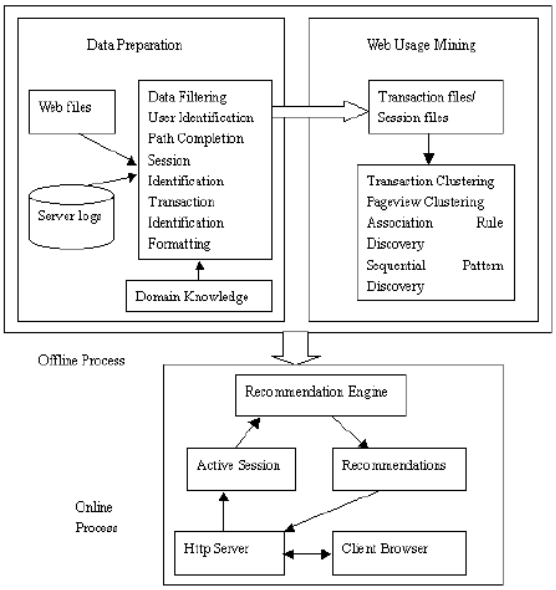

Figure. 1 A system architecture of personalized recommendation using CF based on WUM

3.1 Random forest (RF) classifier

Random Forest (RF) [17] is most accurate ensemble classifier and works efficiently on huge dataset. It can effectively predict the missing data accurately, even in situations where large portions of data are missing and without pre-processing. It combines bagging and random feature selection. RF contains decision trees that are combined individual learners. Random subset of training data is used to generate trees. The test rows are passed through the forest after the forest have been trained. Each tree generates an output class it takes the mode of that classes as the output of RF.

Algorithm for Construction of RF is [18]:

Step 1: Let the number of training cases be "n" and let the number of variables included in the classifier be "m".

Step 2: Let the number of input variables used to make decision at the node of a tree be "p". It assumes that p is always less than "m".

Step 3: Choose a training set for the decision tree by choosing k times with replacement from all "n" available training cases by taking a bootstrap sample. Bootstrapping computes for a given set of data the accuracy in terms of deviation from the mean data. It is usually used for hypothesis tests. Simple block bootstrap can be used when the data can be divided into non-overlapping blocks. But moving block bootstrap is used when it divides the data into overlapping blocks where the portion "k" of overlap between first and second block is always equal to the "k" overlap between second and third overlap and so on. It uses the remaining cases to estimate the error of the tree. Bootstrapping is also used for estimating the properties of the given training data.

Step 4: For each node of the tree, randomly choose variables on which to search for the best split. New data can be predicted by considering the majority votes in the tree. Predict data which is not in the bootstrap sample. And compute the aggregate.

Step 5: Calculate the best split based on these chosen variables in the training set. Base the decision at that node using the best split.

Step 6: Each tree is fully grown and not pruned. Pruning is used to cut of the leaf nodes so that the tree can grow further. Here the tree is completely retained.

Step 7: The best split is one with the least error i.e., the least deviation from the observed data set.

3.2 XGBoost algorithm

The XGBoost algorithm uses a boosting method. The boosting method regards the results obtained by multiple weak classifiers as continuous values, and

these continuous values can be regarded as the value of the loss function, so that weak classification can be used for iterative training to achieve the optimization model effect. The objective function of the XGBoost algorithm, that is, the algorithm loss function, constructs an optimization model by constructing a minimized loss function. The XGBoost model contains multiple CART trees, so the objective function of the model Eq. (1) [19]:

$$O_j(\theta) = \sum_{i=1}^{n} L\left(y_i, \hat{y_l}\right) + \sum_{k=1}^{K} \omega(f_k) \quad (1)$$

Where K is number of trees, f is the functional space of F, F is the set of possible CARTs. Regularization of the objective function Eqs. (2) and (3):

$$L(\phi) = \sum_{i=1}^{n} L\left(y_i, \hat{y_l}\right) + \sum_{k=1}^{K} \omega(f_k) \quad (2)$$

$$\omega(f) = \alpha T + \frac{1}{2}\beta||\omega||^2 \quad (3)$$

T represents the number of nodes in the XGBoost Tree, and ω represents the evaluation results of each node on the product.

The final loss function is (4):

$$L(\phi) = \sum_{i=1}^{n} L\left(y_i, \hat{y_l}\right) + \alpha T + \frac{1}{2}\beta \sum_{t=1}^{T} \omega_{t,i^2} \quad (4)$$

3.3 Shuffled frog leaping algorithm (SFLA)

The SFLA is a meta-heuristic optimization method which is based on observing, imitating, and modeling the behavior of a group of frogs when searching for the location that has the maximum amount of available food. SFLA, originally developed by Eusuff and Lansey in 2003, can be used to solve many complex optimization problems, which are nonlinear, non-differentiable, and multi-modal. SFLA has been successfully applied to several engineering optimization problems such as water resource distribution, bridge deck repairs, job-shop scheduling arrangement, and Traveling Salesman Problem (TSP). The most distinguished benefit of SFLA is its fast convergence speed. The SFLA combines the benefits of the both the genetic-based Memetic Algorithm (MA) and the social behavior-based PSO algorithm [20].

SFLA is a population based random search algorithm inspired by nature memetics. In the SFLA, a population of possible solution defined by a group of frogs that is partitioned into several communities referred to as memeplexes. Each frog in the memeplexes is performing a local search. Within each memeplex, the individual frog's behavior can be influenced by behaviors of other frogs, and it will evolve through a process of memetic evolution. After a certain number of memetics evolution steps, the memeplexes are forced to mix together and new memeplexes are formed through a shuffling process. The local search and the shuffling processes continue until convergence criteria are satisfied.

The SFLA algorithm is [21]:

$Begin;$
$Generate\ random\ population\ of\ P\ solutions\ (frogs);$
$For\ each\ individual\ i \in P\ calculate\ fitness\ (i);$
$Sort\ the\ population\ P\ in\ descending\ order\ of\ \}\ their\ fitness;$
$Divide\ P\ int o\ m\ memeplexes;$
$For\ each\ memeplex\ det e r min e\ the\ best\ and\ worst\ frogs;$
$Im p rove\ the\ worst\ frog\ position\ u sin g\ equations\ (5) or\ (6);$
$Re p eat\ for\ a\ specific\ number\ of\ iterations;$
$End;$
$Combine\ the\ evolved\ memeplexes;$
$Sort\ the\ population\ P\ in\ descending\ order\ of\ their\ fitness;$
$Check\ if\ ter min a tion\ =\ true;$
$End;$

The various steps are as follows:

(1) The SFLA involves a population 'P' of possible solution, defined by a group of virtual frogs (n).

(2) Frogs are sorted in descending order according to their fitness and then partitioned into subsets called as memeplexes (m).

(3) Frogs i is expressed as $X_i = (x_{i1}, x_{i2}, \ldots, x_{iS})$ where S represents number of variables.

(4) Within each memeplex, the frog with worst and best fitness are identified as Xw and Xb.

(5) Frog with global best fitness is identified as Xg.

(6) The frog with worst fitness is improved according to the following Eqs. (5) and (6).

$$D_i = rand()\ (X_b - X_w) \quad (5)$$

$$X_{neww} = X_{oldw} + D_i(-Di_{max_{max}} \quad (6)$$

Where rand is a random number in the range of [0, 1], Di is the frog leaping step size of the i-th frog and Dmax is the maximum step allowed change in a frog's position. If the fitness value of new Xw is better than the current one, Xw will be accepted. If it isn't improved, then the calculated (1) and (2) are repeated with Xb replaced by Xg. If no improvement becomes possible in the case, a new Xw will be

generated randomly. Repeat the update operation for a specific number of iterations.

After a predefined number of memetic evolutionary steps within each memeplex, the solutions of evolved memeplexes are replaced into new population. This is called the shuffling process. The shuffling process promotes a global information exchange among the frogs. Then, the population is sorted in order of decreasing performance value and updates the population best frog's position, repartition the frog group into memeplexes, and progress the evolution within each memeplex until the conversion criteria are satisfied.

3.4 Proposed SFLA with RF algorithm

In order to improve the accuracy of the model in recommender system, this work propose a new combination algorithm, namely: SFLA-based RF (SFLA-RF), whose principle is to use the SFLA to assign a weight to each decision tree in the RF, so that the sub-tree with high accuracy has a higher weight, and the weight is depended on the performance of the sub-tree.

The main idea of SFLA-RF is to assign a weigh value to each tree in RF, and the values are searched for the optimal solution by SFLA. Its establishment steps are as follows [22]:

Step 1: After m times sampling with replacement from the original data set, a new data set with m samples (there may be duplicate samples) can be obtained. Also, using the rule of sampling without replacement, f features are taken from the n features as input features.

Step 2: For the new sample-set D (with m samples and f features), if the subset of the samples belonging to class ck is Ck, then the Gini impurity is Eq. (7):

$$Gini(D) = 1 - \sum_{k=1}^{K} \left(\frac{|C_k|}{|D|}\right)^2 \quad (7)$$

For each feature A and its possible value, a, calculate Gini (D, A) according to (8):

$$\begin{gathered} Gini(D,A) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2) \\ D_1 = \left\{(\vec{x}, y) \in D | \vec{x}^{(A)} = a\right\} \\ D_2 = \left\{(\vec{x}, y) \in D | \vec{x}^{(A)} \neq a\right\} = D - D_1 \end{gathered} \quad (8)$$

Step 3: Selection of optimal feature and optimal segmentation point: The A and a, which minimize Gini impurity, are the optimal feature and optimal segmentation point. According to them, the training set is divided into two sub-nodes.

Step 4: Recursively call the step 2 and step 3 for these two sub-nodes. And finally, a decision tree is constructed with the new data set (with m samples and f features).

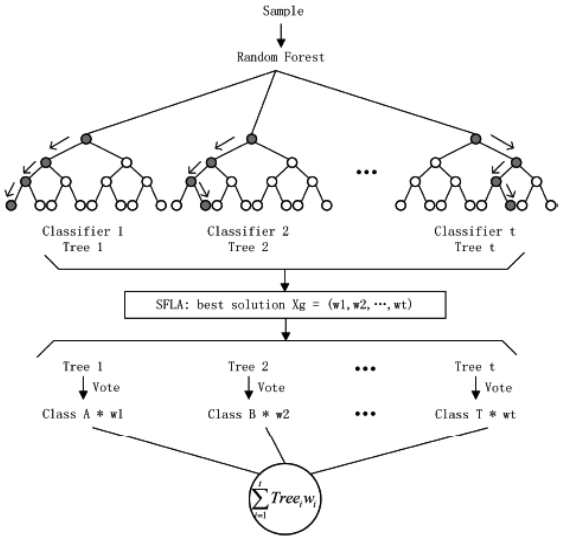

Figure. 2 Structure of SFLA-RF model

Step 5: Repeat step1 to step 4 t times to construct t decision trees to form a RF model.

Step 6: Initialize SFLA parameters.

Step 7: Generate a frog population.

Step 8: Divide frogs into memeplexes.

Step 9: Memetic evolution.

Step 10: Shuffle memeplexes.

Step 11: If the number of global searches is less than its maximum: t + 1 < Gmax, then jump to the step 8. Or, output the best frog as the best solution.

The structure of SFLAL-RF is shown in Fig. 2.

SFLA-RF is an ensemble learner composed of t base learners: $tree_1, tree_2, \ldots, tree_t$, and the final decision result is Eq. (9):

$$SFLA - RF(\vec{x}) = X_g.RF = \sum_{i=1}^{t} tree_i(\vec{x}) w_i \quad (9)$$

SFLA algorithm combines the advantages of two population intelligence optimization algorithms, namely, meme evolution based memetic algorithm and swarm behavior-based Particle Swarm Optimization (PSO) algorithm. Some traditional bionic algorithms such as genetic algorithm and PSO have poor global convergence, whereas SFLA is a global convergence algorithm. In theory, as long as the number of iterations meets the requirements, SFLA will find the global optimal solution.

The structure of SFLAL-RF is shown in Fig. 2.

3.5 Proposed SFLA with XGBoost algorithm

XGBoost, as an excellent machine learning algorithm in recent years, has good running speed and

accuracy and is widely used in classification problems. When using XGBoost classification, it is necessary to adjust the parameters of the trainer to improve its performance. The choice of parameters determines the accuracy of the XGBoost model [23]. The commonly used parameter adjustment method is generally the grid search method, but the search range of the grid search method is too narrow, and it is not easy to find the optimal parameters. This work proposes an XGBoost classification algorithm based on SFLA optimization parameters. The Python toolkit XGBoost is selected to optimize three important parameters in the XGBoost classifier: learning rate (learning_rate, ETA for short), maximum depth of the tree (max_depth), and sample sampling rate (subsample).

Learning_rate: when updating leaf nodes, the weight will be multiplied by ETA. By reducing the weight of the feature, the promotion calculation process is more conservative. +e commonly used value range is [0, 1], and the default value is 0.3.

Max_depth: it controls the complexity of the decision tree. The larger the value, the more complex the model, but over fitting will occur. The default value is 6.

Subsample: the subsample ratio of the training set means that XGBoost selects the sample ratio of the first spanning tree, which can effectively prevent over fitting. The default value is 1.

The proposed method based on SFLA is detailed as follows:

Step 1: preprocess the collected data with a normalized method.

Step 2: the preprocessed data are subjected to CF-WUM so as to facilitate subsequent training of the XGBoost model.

Step 3: initialize the SFLA, where the initial parameters of the algorithm are given. Set each frog in a 3-dimensional space, and encode the three dimensions as key parameters ETA, max_depth, and subsample, respectively.

Step 4: set the upper and lower limits of the XGBoost algorithm parameters that need to be optimized to generate the initial population of frogs so that the position of each frog is within a suitable range.

Step 5: according to the obtained frog population, based on the XGBoost model, calculate the fitness of each frog position.

Step 6: sort the obtained fitness values to get the best frog position of the current frog population and save it as the current global best position.

Step 7: update the position of the frog in each subgroup through Eqs. (5) and (6).

Step 8: enter iterative optimization and repeat Steps 3–5. When the number of iterations reaches the maximum, stop the loop and obtain the best parameters ETA, max_depth, and subsample from the final best frog position.

Step 9: bring the obtained best parameters ETA, max_depth, and subsample into the XGBoost model to get the best frog position after training.

4. Results and discussion

The experiments were conducted for CF, WUM, Hybrid CF-WUM with RF, XGBoost, proposed SFLA-RF and proposed SFLA-XGBoost. The methods are evaluated for two scenarios. When the number of rating is less than 15 and when the number of ratings are more than 15 to check the efficacy of the methods when sparse data is available. The performance metrics used for evaluation are sensitivity, Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). Table 1 to 6 and Fig. 3 to 8 shows the results obtained.

Table 1. Average sensitivity for rating <=15

	CF	WUM	Hybrid CF-WUM
RF	0.3516	0.3903	0.4518
XGBoost	0.3659	0.4059	0.4699
Proposed SFLA RF	0.3768	0.4202	0.486
Proposed SFLA XGBoost	0.3945	0.4334	0.5074

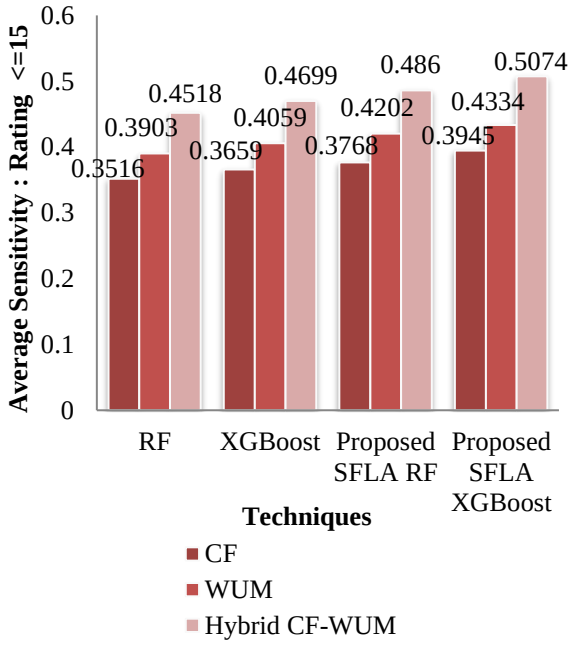

Figure. 3 Average sensitivity for rating <=15

From the Fig. 3, it can be observed that the proposed SFLA-XGBoost has higher average sensitivity: rating <=15 for CF by 11.49% for RF, by 7.52% for XGBoost and by 4.58% for proposed SFLA-RF respectively. The proposed SFLA-XGBoost has higher average sensitivity: rating <=15 for WUM by 10.46% for RF, by 6.55% for XGBoost and by 3.09% for proposed SFLA-RF respectively. The proposed SFLA-XGBoost has higher average sensitivity: rating <=15 for hybrid CF-WUM by 11.59% for RF, by 7.67% for XGBoost and by 4.31% for proposed SFLA-RF respectively.

From the Fig. 4, it can be observed that the proposed SFLA-XGBoost has higher average sensitivity: rating >15 for CF by 10.78% for RF, by 7.23% for XGBoost and by 3.44% for proposed SFLA-RF respectively. The proposed SFLA-XGBoost has higher average sensitivity: rating >15 for WUM by 11.36% for RF, by 6.41% for XGBoost and by 4.13% for proposed SFLA-RF respectively. The proposed SFLA-XGBoost has higher average

Table 2. Average sensitivity for rating >15%

	CF	WUM	Hybrid CF-WUM
RF	0.3818	0.4209	0.4888
XGBoost	0.3956	0.4423	0.5052
Proposed SFLA RF	0.4109	0.4525	0.5293
Proposed SFLA XGBoost	0.4253	0.4716	0.5488

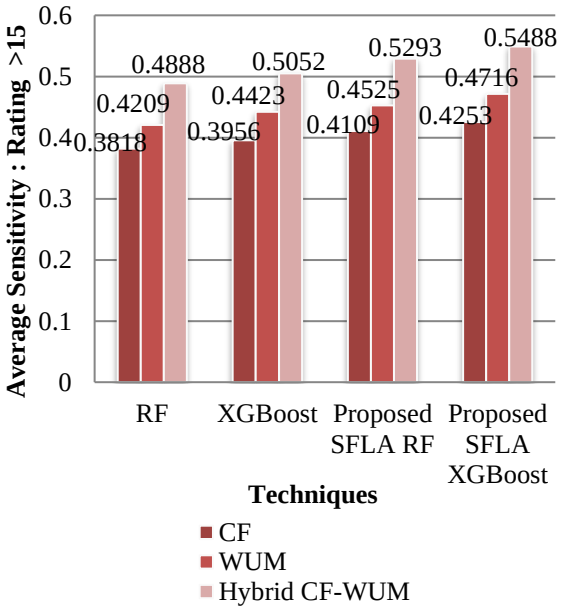

Figure. 4 Average sensitivity for rating >15

Table 3. Average root mean square error (RMSE) for rating <=15

	CF	WUM	Hybrid CF-WUM
RF	0.8301	0.8327	0.821
XGBoost	0.7948	0.8001	0.787
Proposed SFLA RF	0.7598	0.7604	0.7481
Proposed SFLA XGBoost	0.7325	0.7291	0.7124

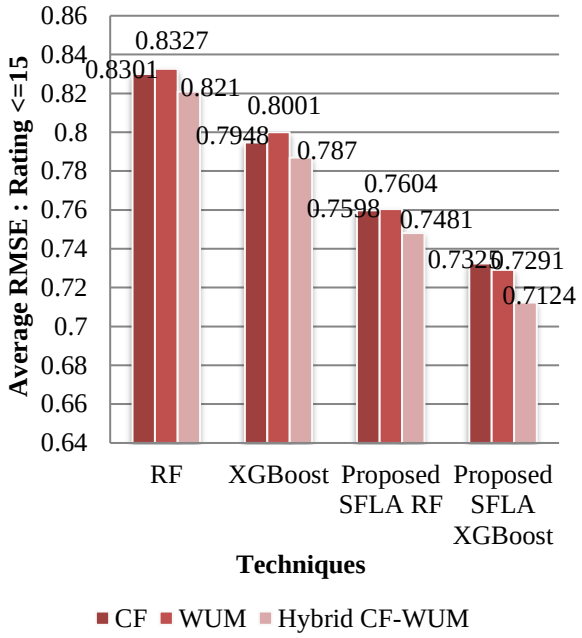

Figure. 5 Average root mean square error (RMSE) for rating <=15

sensitivity: rating >5 for hybrid CF-WUM by 11.56% for RF, by 8.27% for XGBoost and by 3.62% for proposed SFLA-RF respectively.

From the Fig. 5, it can be observed that the proposed SFLA-XGBoost has lower average RMSE: rating <=15 for CF by 12.49% for RF, by 8.15% for XGBoost and by 3.66% for proposed SFLA-RF respectively. The proposed SFLA-XGBoost has lower average RMSE: rating <=15 for WUM by 13.27% for RF, by 9.28% for XGBoost and by 4.2% for proposed SFLA-RF respectively. The proposed SFLA-XGBoost has lower average RMSE: rating <=15 for hybrid CF-WUM by 14.16% for RF, by 9.95% for XGBoost and by 4.88% for proposed SFLA-RF respectively.

From the Fig. 6, it can be observed that the proposed SFLA-XGBoost has lower average RMSE: rating >15 for CF by 12.16% for RF, by 8.52% for

Table 4. Average root mean square error (RMSE) for rating >15

	CF	WUM	Hybrid CF-WUM
RF	0.7813	0.7894	0.7785
XGBoost	0.7533	0.7555	0.7422
Proposed SFLA RF	0.7163	0.7175	0.705
Proposed SFLA XGBoost	0.6917	0.6868	0.6735

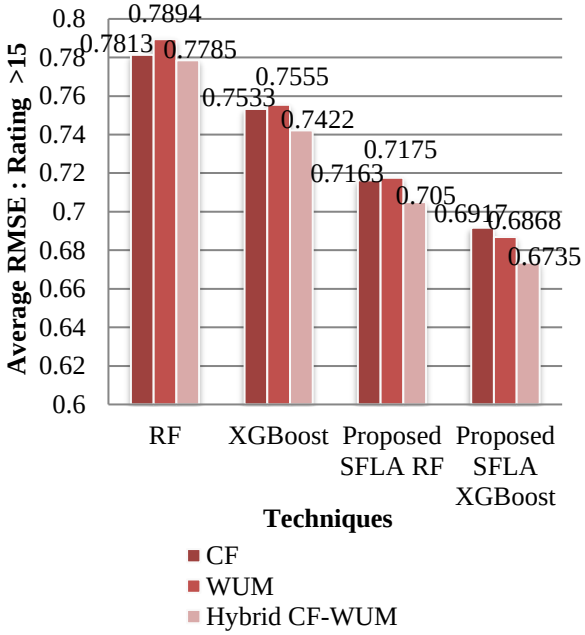

Figure. 6 Average root mean square error (RMSE) for rating >15

Table 5. Average mean absolute error (MAE) for rating <=15

	CF	WUM	Hybrid CF-WUM
RF	0.849	0.818	0.8451
XGBoost	0.8033	0.8201	0.7846
Proposed SFLA RF	0.778	0.7659	0.7423
Proposed SFLA XGBoost	0.7266	0.7147	0.7011

XGBoost and by 3.49% for proposed SFLA-RF respectively. The proposed SFLA-XGBoost has lower average RMSE: rating >15 for WUM by 13.9% for RF, by 9.52% for XGBoost and by 4.37% for proposed SFLA-RF respectively. The proposed SFLA-XGBoost has lower average RMSE: rating >15 for hybrid CF-WUM by 14.46% for RF, by 9.7%

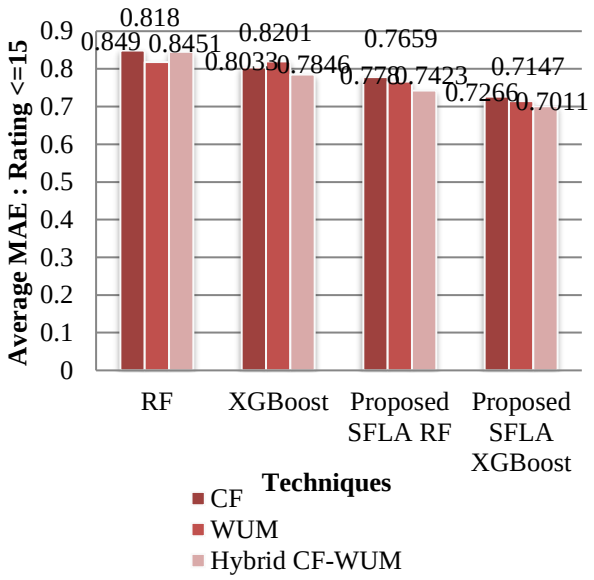

Figure. 7 Average mean absolute error (MAE) for rating <=15

Table 6. Average MAE: mean absolute error (MAE) for rating >15

	CF	WUM	Hybrid CF-WUM
RF	0.7715	0.8059	0.7849
XGBoost	0.774	0.7481	0.723
Proposed SFLA RF	0.6987	0.7298	0.7251
Proposed SFLA XGBoost	0.701	0.7015	0.6546

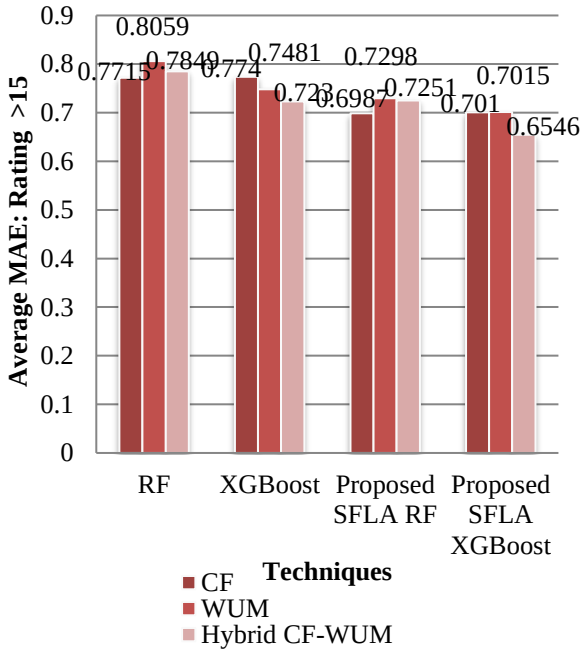

Figure. 8 Average MAE: mean absolute error (MAE) for rating >15

for XGBoost and by 4.57% for proposed SFLA-RF respectively.

From the Fig. 7, it can be observed that the proposed SFLA-XGBoost has lower average MAE: rating <=15 for CF by 15.53% for RF, by 10.03% for XGBoost and by 6.83% for proposed SFLA-RF respectively. The proposed SFLA-XGBoost has lower average MAE: rating <=15 for WUM by 13.47% for RF, by 13.73% for XGBoost and by 6.92% for proposed SFLA-RF respectively. The proposed SFLA-XGBoost has lower average MAE: rating <=15 for hybrid CF-WUM by 18.63% for RF, by 11.24% for XGBoost and by 5.71% for proposed SFLA-RF respectively.

From the Fig. 8, it can be observed that the proposed SFLA-XGBoost has lower average MAE: rating >15 for CF by 9.57% for RF, by 9.89% for XGBoost and by 0.33% for proposed SFLA-RF respectively.

The proposed SFLA-XGBoost has lower average MAE: rating >15 for WUM by 13.85% for RF, by 6.43% for XGBoost and by 3.95% for proposed SFLA-RF respectively. The proposed SFLA-XGBoost has lower average MAE: rating >15 for hybrid CF-WUM by 18.1% for RF, by 9.93% for XGBoost and by 10.22% for proposed SFLA-RF respectively

5. Conclusion

The development and popularization of E-commerce, more and more information services have appeared on the web. In order to meet users requirements more accurately, several personalized recommendation systems had been set up. Many methods have been proposed to discover users' interests for service recommendation, such as CF and content based recommendation. In this work, a new personalized recommendation method is proposed based on user's interest, which combines CF based on WUM, RF and XGBoost. RF are collection of ensembles of decision trees, used for prediction on basis of some predictor values. XGBoost algorithm improves the calculation method of the objective function on the basis of gradient boosting and reduces the calculation time. Therefore, this work proposes an CF-WUM algorithm based on the SFLA optimized RF and the SFLA optimized XGBoost model, which uses the powerful optimization capabilities of the SFLA algorithm to optimize key parameters for RF and XGBoost, effectively improving the prediction accuracy of the RF and XGBoost model, so as to more accurately get best frogs and best position. Results show that the proposed SFLA-XGBoost has higher average sensitivity: rating <=15 for CF by 11.49% for RF, by 7.52% for XGBoost and by 4.58% for proposed SFLA-RF respectively. The proposed SFLA-XGBoost has higher average sensitivity: rating <=15 for WUM by 10.46% for RF, by 6.55% for XGBoost and by 3.09% for proposed SFLA-RF respectively. The proposed SFLA-XGBoost has higher average sensitivity: rating <=15 for hybrid CF-WUM by 11.59% for RF, by 7.67% for XGBoost and by 4.31% for proposed SFLA-RF respectively.

Appendix

Notation List

K -number of trees
F - set of possible CARTs.
f - functional space of *F*
T - number of nodes in the XGBoost Tree
ω - evaluation results of each node on the product
n -virtual frogs
P - population of possible solution in SFLA
Xw - frog with worst fitness
Xb. - frog with best fitness
Xg - frog with global best fitness
rand - random number in the range of [0, 1]
Di - frog leaping step size of the i-th frog
Dmax- maximum step allowed change in a frog's position.
Gmax - number of global searches.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, 1 and 2; methodology, 1; software, 1; validation, 1 and 2; formal analysis, 1; investigation, 1; resources, 1; writing—original draft preparation, 1; writing—review and editing, 1; supervision, 2; project administration, 2;

Reference

[1] E. Afzalan, M. A. Taghikhani, and M. Sedighizadeh, "Optimal placement and sizing of DG in radial distribution networks using SFLA", *International Journal of Energy Engineering*, Vol. 2, No. 3, pp. 73-77, 2012.

[2] A. Ajesh, J. Nair, and P. S. Jijin, "A random forest approach for rating-based recommender system", In: *Proc. of International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1293-1297, 2016.

[3] B. K. Poornima, D. Deenadayalan, and A. Kangaiammal, "Text preprocessing on extracted text from audio/video using R", *International*

Journal of Computational Intelligence and Informatics, Vol. 6, No. 4, pp. 267-278, 2017.

[4] B. Chang, R. Yang, C. Guo, S. Ge, and L. Li, "A new application of optimized random forest algorithms", *Intelligent Fault Location of Rudders*, Vol. 7, No. 5, pp. 94276-94283. 2019.

[5] D. Deenadayalan, A. Kangaiammal, and B. K. Poornima, "Learner Level and Preference Prediction of E-learners for E-learning Recommender Systems", *Studies in Computational Intelligence- Springer*, Vol. 771, pp. 133-139, 2019.

[6] D. Deenadayalan, A. Kangaiammal, and B. K. Poornima, "EEG based learner's learning style and preference prediction for E-learning", In: *Proc. of 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, pp. 316-320, 2018.

[7] Q. Du, N. Li, S. Yang, D. Sun, and W. Liu, "Integrating KNN and Gradient Boosting Decision Tree for Recommendation", In: *Proc. of IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pp. 2042-2049, 2021.

[8] R. Duan, C. Jiang, and H. K. Jain, "Combining review-based collaborative filtering and matrix factorization: A solution to rating's sparsity problem", *Decision Support Systems*, Vol. 156, pp. 736-748, 2022.

[9] F. H. Nanehkaran, S. M. Lajevardi, and M. M. Bidgholi, "Nearest neighbours algorithm and genetic-based collaborative filtering", *Concurrency and Computation: Practice and Experience*, Vol. 34, No. 1, pp. 38-65, 2022.

[10] F. H. Nanehkaran, S. M. Lajevardi, and M. M. Bidgholi, "Optimization of fuzzy similarity by genetic algorithm in user-based collaborative filtering recommender systems", *Expert Systems*, Vol. 39, No. 4, p. e12893, 2022.

[11] S. Jiang, J. Ding, and L. Zhang, "A Personalized Recommendation Algorithm Based on Weighted Information Entropy and Particle Swarm Optimization", *Mobile Information Systems*, Vol. 2021, pp. 96-104, 2021.

[12] A. Kangaiammal and B. K. Poornima, "Personalization of Learning Objects in E-learning System using David Merrill's approach with Web 3.0.", *International Journal of Applied Engineering Research (IJAER)*, Vol. 10, No. 85, pp. 318-324, 2015.

[13] J. Li and Z. Ye, "Course recommendations in online education based on collaborative filtering recommendation algorithm", *Complexity*, Vol. 2020, pp. 156-165, 2020.

[14] J. Li, J. Xu, and M. Yang, "Collaborative filtering recommendation algorithm based on KNN and Xgboost hybrid", *Journal of Physics: Conference Series*, Vol. 1748, No. 3, pp. 032-041, 2021.

[15] H. Liu, Z. Hu, A. Mian, H. Tian, and X. Zhu, "A new user similarity model to improve the accuracy of collaborative filtering", *Knowledge-Based Systems*, Vol. 56, No. 7, pp. 156-166, 2014.

[16] T. R. Prajwala, "A comparative study on decision tree and random forest using R tool", *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 4, No. 1, pp. 196-199. 2015.

[17] T. Puraram, P. Chaovalit, A. Peethong, P. Tiyanunti, S. Charoensiriwath, and W. Kimpan, "Thai Food Recommendation System using Hybrid of Particle Swarm Optimization and K-Means Algorithm", In: *Proc. of 6th International Conference on Machine Learning Technologie*, pp. 90-95, 2021.

[18] G. G. Samuel and C. C. A. Rajan, "A Modified Shuffled Frog Leaping Algorithm for Long-Term Generation Maintenance Scheduling", In: *Proc. of the Third International Conference on Soft Computing for Problem Solving*, pp. 11-24, 2014.

[19] Y. S. Sneha, G. Mahadevan, and M. Prakash, "A personalized product based recommendation system using web usage mining and semantic web", *International Journal of Computer Theory and Engineering*, Vol. 4, No. 2, pp. 202-205, 2012.

[20] Y. Song, H. Li, P. Xu, and D. Liu, "A method of intrusion detection based on woa-xgboost algorithm", *Discrete Dynamics in Nature and Society*, Vol. 2022, No. 7, pp. 321-330, 2022.

[21] T. A. Syed and S. S. K. Nair, "Personalized recommendation system for advanced learning management systems", In: *Proc. of the 8th International Conference on Information Communication and Management*, pp. 90-95, 2018.

[22] T. A. Syed, V. Palade, R. Iqbal, and S. S. K. Nair, "A Personalized Learning Recommendation System Architecture for Learning Management System", In: *Proc. of the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (KDIR 2017)*, pp. 275-282. 2017.

[23] T. Wang and Y. Ren, "Research on personalized recommendation based on web usage mining using collaborative filtering technique", *WSEAS*

Transactions on Information Science and Applications, Vol. 6, No. 1, pp. 62-72. 2009.

[24] X. Wang, F. Luo, C. Sang, J. Zeng, and S. Hirokawa, "Personalized movie recommendation system based on support vector machine and improved particle swarm optimization", *IEICE Transactions on Information and Systems*, Vol. 100, No. 2, pp. 285-293. 2017.

[25] G. Xu, Z. Tang, C. Ma, Y. Liu, and M. Daneshmand, "A collaborative filtering recommendation algorithm based on user confidence and time context", *Journal of Electrical and Computer Engineering*, Vol. 2019, No. 7, pp. 326-337, 2019.

[26] W. Yue, Z. Wang, W. Liu, B. Tian, S. Lauria, and X. Liu, "An optimally weighted user-and item-based collaborative filtering approach to predicting baseline data for Friedreich's Ataxia patients", *Neurocomputing*, Vol. 419, pp. 287-294, 2021.

[27] S. Zubair and M. A. A. Sabri, "Hybrid Measurement of the similarity value based on a Genetic Algorithm to improve prediction in a collaborative filtering recommendation system", *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, Vol. 10, No. 2, pp. 165-182, 2021.

[28] P. K. Balasamy and K Athiyappagounder, "An optimized Feature Selection Method for E-Learning Recommender System Using Deep Neural Network Based on Multilayer perceptron", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 5, 2022, doi: 10.22266/ijies2022.1031.40.