

## Hybrid Recommender System Using Systolic Tree for Pattern Mining

S. Rajalakshmi<sup>1,\*</sup> and K. R. Santha<sup>2</sup>

<sup>1</sup>Department of Computer Science Engineering, Sri Venkateswara College of Engineering, Sriperumbudur, India

<sup>2</sup>Department of Electrical and Electronics Engineering, Sri Venkateswara College of Engineering, Sriperumbudur, India

\*Corresponding Author: S. Rajalakshmi. Email: rajiaslakshmi@gmail.com

Received: 01 October 2021; Accepted: 16 February 2022

**Abstract:** A recommender system is an approach performed by e-commerce for increasing smooth users' experience. Sequential pattern mining is a technique of data mining used to identify the co-occurrence relationships by taking into account the order of transactions. This work will present the implementation of sequence pattern mining for recommender systems within the domain of e-commerce. This work will execute the Systolic tree algorithm for mining the frequent patterns to yield feasible rules for the recommender system. The feature selection's objective is to pick a feature subset having the least feature similarity as well as highest relevancy with the target class. This will mitigate the feature vector's dimensionality by eliminating redundant, irrelevant, or noisy data. This work presents a new hybrid recommender system based on optimized feature selection and systolic tree. The features were extracted using Term Frequency-Inverse Document Frequency (TF-IDF), feature selection with the utilization of River Formation Dynamics (RFD), and the Particle Swarm Optimization (PSO) algorithm. The systolic tree is used for pattern mining, and based on this, the recommendations are given. The proposed methods were evaluated using the MovieLens dataset, and the experimental outcomes confirmed the efficiency of the techniques. It was observed that the RFD feature selection with systolic tree frequent pattern mining with collaborative filtering, the precision of 0.89 was achieved.

**Keywords:** Recommender systems; hybrid recommender systems; frequent pattern mining; collaborative filtering; systolic tree; river formation dynamics; particle swarm optimization

### 1 Introduction

Of late, there has been an increase in the volume of digital information, online services, and electronic sources. However, information overload leads to a potential issue of filtering as well as effectively delivering the relevant information to a user. Moreover, there must be prioritization of the information for a user instead of only obtaining the required information, which can result in issues of information overload. Even though search engines filter the web pages to match the queries of the Internet users, it is extremely challenging to identify a user's requirements by utilizing simple keywords. While the Semantic Web through intelligent



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

search queries does offer some assistance in detecting pertinent information, unfortunately, it is dependent on the degree of the webpages' annotation [1].

The issue emphasizes the necessity of information extraction systems that can filter hidden information as well as envisage a user's preference for a given source. These systems are referred to as recommender systems, and they can significantly mitigate the aforementioned problems. For a given new item, the recommender systems can predict with higher accuracy the likes and dislikes of the user to a particular item, based on the user preferences, observed behavior, and the item's related information (content or demographic information).

Various literature has diverse ways of classifying recommender systems. These systems are classified into four distinct types following the methods employed: content, knowledge, Collaborative Filtering (CF), and hybrid-based recommender systems. In a content-based recommender system, recommendations are based on the item content earlier preferred by the user. Typically, there is the application of a heuristic method for matching the item attribute based on the keyword to the user's profile. The content-based is generally utilized in either book or web page recommendations wherein much significance is given to the content's semantic meaning [2].

In a knowledge-based recommender system, the relationship between the user's requirement and the item's function is used for recommendation. This method is dependent on the existing knowledge or rules wherein there is a requirement for knowledge database management. In a CF-based recommender system, users sharing similar tastes are considered for making recommendations. A user-item rating matrix is created and a popularly employed technique in e-commerce. Despite this method's drawbacks, such as the rating matrix's sparsity and the scalability when handling huge data amounts, it does have a unique position in the recommender system's development. In a hybrid recommendation system [3], at least two types of recommendation strategies are employed to increase the performance and mitigate the shortcomings of each strategy.

Recommender systems have to confront numerous challenges as well as problems. In particular, these issues are observed in the CF systems, the most effective and precise approach. For example, based on past user behavior, the CF systems have to collect certain information related to the visitors' preferences. This stored information is termed a user model or a user profile. Furthermore, yet another problem will occur when new items are included. In the CF method's case, it is yet to be assigned to any user and cannot be recommended to anyone. The content-based approach resolves this issue by evaluating the similarity between the new and existing items [4].

In an arbitrary recommender system application, while there are many items on offer, a user can visit only a very limited number of items during a single session. It will result in the sparsity of the input data as well as lower reliability with regards to gauging the similarity between customers. Despite all this, scalability is ultimately the most critical challenge in the online recommendations field. Even though the recommender system has to handle huge amounts of dynamic data, there must be a reasonable time for result generation to employ them in real-time applications. The user's expectation on seeing news or item will expect to see recommendations immediately, while analysis is required for millions of archived news. Furthermore, there are hybrid approaches that involve the combination of at least two distinct methods. These approaches will try to resolve the problems in each method by using the other one's advantages.

The data mining procedure's [5] objective is to identify patterns from the users' activities which helps in generating recommendations. The sequential pattern mining's input will be a data sequence that has been pre-processed in the earlier steps, while its output will be a set of frequent patterns. Frequent patterns [6] are used to yield the rules for a recommendation. Every rule will constitute predictions of particular patterns dependent on the existing patterns. Antecedent  $\rightarrow$  consequent will indicate the form of the rules, where the previous pattern is the antecedent. The patterns' confidence should fulfill the minimum confidence for

becoming a rule. The recommender system will compare the users' patterns against the rules' antecedent. If there is a match of the antecedent with a users' pattern, the system will recommend the rules' consequent to the users. Sorting is done on the rules based on the confidence value, and items with higher confidence are given as recommendations by the recommender system.

Feature selection is a computational method for resolving the issue of picking features based on a definition of an evaluation criterion or relevance. The adopted relevance strongly connects with the final objective [7]. The feature selection methods are categorized as filter, wrapper, and hybrid methods based on the evaluation criteria. Feature subset selection is treated as a Non-deterministic Polynomial (NP)-hard problem. To put it simply, detection of the best subset is done by using a comprehensive search strategy for assessing all the potential subsets. While this method ensures an optimal feature subset, it is extremely time-consuming and impractical even for datasets of modest size. As the evaluation of all potential subsets is an expensive task, the search must be done on an acceptable feature subset regarding both appropriateness and computational complexity. An approach for resolving NP-Hard problems is the meta-heuristic algorithm. These algorithms are approximate methods that can identify satisfactory solutions over acceptable periods rather than identifying the optimal solution. Meta-heuristic algorithms belong to the category of approximate optimization algorithms, which strategies for avoiding the local optima and have applications in an extensive range of optimization problems [8].

Multiple feature selection methods will employ the meta-heuristics to avoid the high dimensional dataset's increasing computational complexity. These algorithms will employ primitive mechanisms as well as operations for resolving an optimization problem and also for the optimal solution's iterative search. Evolutionary Algorithms (EA) and Swarm Intelligence (SI) are the two distinct classifications of meta-heuristic approaches. This work has given the proposal for feature selection based on the River Formation Dynamics (RFD) and the Particle Swarm Optimization (PSO) algorithm for a hybrid recommender system-based pattern mining. The main contributions of this work are the use of a systolic tree for finding recommendations and the use of RFD for feature selection. The proposed recommender system is a hybrid recommender system that uses content and is based on collaborative filtering to generate certain recommendations for users. The recommender system is useful for finding recommendations in e-commerce sites for gadgets, books, home appliances, computers, fashion, and lifestyle. Also, for sites related to services, travels, movies, books, and so on.

The rest of this investigation has been arranged into the subsequent sections. Section two will elaborate on the related works available in the literature. Section three will describe this work's numerous techniques used in this investigation. Section four presents the experimental outcomes, and Section five offers the work's conclusions.

## 2 Related Works

Bobadilla et al. [9] had offered a deep learning-based method, known as the DeepUnHide, which could extract demographic data from the users as well as the items factors in the CF recommender systems. This proposed method's key principle was on the gradient-based localization, which was utilized in the image processing literature for highlighting each classification class's representative areas. Validation experiments had employed two public datasets as well as current baselines. The experimental outcomes had shown DeepUnHide's superiority in making feature selection and demographic classification in comparison to advanced methods of feature selection.

Fang et al. [10] had focused on matrix factorization-based recommender systems due to their extensive industrial implementations. For a given number of fake users injected by the attacker, the authors formulated the crafting of the fake users' rating scores as an optimization problem. Even so, the resolution of the optimization problem proved to be quite challenging since it was a non-convex integer programming

problem. For addressing this challenge, the authors had devised numerous techniques for the approximate resolution of the optimization problem. As an example, the authors had leveraged the influence function for identifying normal users who were influential in making recommendations and also for resolving the formulated optimization problem based on these influential users. The experimental results had demonstrated that the attacks were effective and had surpassed the performance of other existing methods.

Gogna et al. [11] put forward a single-stage optimization to accomplish high diversity whilst retaining the required accuracy levels. The authors had proposed the incorporation of additional diversity improving constraints in the CF's matrix factorization model. The conventional matrix factorization scheme would yield dense user and item latent factor matrices. This model's concept was driven by the fact that while the user would show a certain likeness towards all the latent factors, an item cannot have all the features; and hence, yield a sparse structure. In addition, the authors had proposed a method for the formulation. The proposed model's advantage over existing advanced techniques was demonstrated by the outcomes of the experimentations performed on a real-world movie database.

Bansal et al. [12] had proposed the novel Bi-clustering based Memetic Algorithm for Recommender Systems (Bi-MARS) based on the memes' collaborative behavior. The bi-clusters were formed to discover the target user's precise as well as localized neighborhood. In addition, there was the formulation of a local search for refining the similarity values connected with the neighborhood users and the prediction score function for unrated items. The authors have assessed the performance of the Bi-MARS on the MovieLens dataset of three distinct sizes.

Hamada et al. [13] presented a framework that trained the Artificial Neural Networks (ANN) with the PSO and also had employed the NNs for integration of the multi-criteria rating information as well as for determination of the user preferences. There was the integration of the proposed ANN with the K-Nearest Neighborhood (KNN) CF to predict the unknown criteria ratings. Testing was done on the proposed model with a multi-criteria dataset to recommend movies to the users. This work's empirical results had demonstrated the proposed model's higher prediction accuracy when compared against the corresponding conventional recommendation technique as well as other multi-criteria recommender systems.

Mohammadpour et al. [14] had suggested a single-objective hybrid evolutionary method for clustering the items in the offline CF recommender system. At first, this proposed approach would generate a random set of solutions as a population. Later, at every iteration, this approach would initially enhance the solution population with the utilization of the Genetic Algorithm (GA) and then with the utilization of the Gravitational Emulation Local Search (GELS) algorithm. Experiments indicated that while the proposed hybrid method had needed a relatively high run time, it had resulted in more suitable clustering of the existing data.

Katarya et al. [15] had proposed a hybrid model for recommending movies that utilized the type division method and also categorized the movie types following the users, resulting in mitigation of the computation complexity. K-Means had offered initial parameters to the PSO for boosting its performance. The PSO offered the initial seed and optimized the Fuzzy C-Means (FCM) for the data items' (users) soft clustering rather than strict clustering behavior in K-Means. At first, the authors had used the type division method on the proposed model for the dense multi-dimensional data space's minimization. Upon examination of numerous techniques, the authors found that the FCM could offer better results than that of the K-Means. Since the GA had the limitation of unguided mutation, this work had employed the PSO. Experimentations on the Movie Lens dataset had illustrated that the proposed model was able to offer high performance in terms of veracity and also was able to deliver recommendations that were more predictable as well as personalized.

Shen et al. [16] proposed a deep learning technique for finding the relationship between the description of each listing and its price in online short-term rental platforms, and a text-based price recommendation is

given. The proposed method was called TAPE, which recommends a realistic price for newly added listings. For evaluation, listings in Boston, London, Los Angeles, and New York City were used. The experiments demonstrated that the proposed achieved lower Root-Mean-Square Error (RMSE) when compared to existing methods.

Nitu et al. [17] presented a travel recommendation system using social media posts. Personalized travel recommendations are provided based on user-specific preferences and needs. The Twitter data was analyzed to find the user's preferences, and also the data of their friends and followers were analyzed. Relevant tweets related to travel are identified using a machine learning classifier and used to give personalized travel recommendations. The proposed method also includes time-sensitive recency weight to obtain the user's most recent interest.

### 3 Methodology

Discussions about the systolic tree, the TF-IDF feature extraction method, the PSO, and the RFD-based feature selection methods are given in this section.

#### 3.1 Systolic Tree

A systolic tree is an array of the pipelined Processing Elements (PEs) in a multi-dimensional tree pattern. The aim is to replicate the Systolic tree algorithm's [18] internal memory layout whilst accomplishing higher throughput. The systolic role of the systolic tree as mapped in the Field Programmable Gate Array (FPGA) hardware will be similar to the *U*-Prefix Span as utilized in the software. The direct translation of a software algorithm into a hardware architecture may not always be either practical or efficient. The approach will construct the tree based on the maximum node degree approximation. Though, frequent itemsets may not be found when the actual node degree at a certain point in the tree is more the assessed node degree. If *n* indicates the number of items in the database, *K* will indicate the assessed maximum node degree approximation, and *W* will indicate the systolic tree's assessed depth. In the static tree structure, every node will have *K* children. Eq. (1) will express the tree's total number of nodes as follows:

$$K^W + K^{W-1} + \dots + K^1 = \frac{K(K^W - 1)}{K - 1} \quad (1)$$

When there is a large *K*, each node will have more children, leading to each node requiring a large number of interfaces. As a result, each node's inner structure will end up being very complex. For simplification of the node's complexity, each node will be assigned two interfaces rather than the *K* number of interfaces. While one interface is connected to its first child, the other interface is connected to its closest sibling. For the systolic tree structure, every PE will have a unique parent and hence, will only have a single path that traces back from any PE to the control PE. The dictation's key principle is that any path containing the queried candidate itemset will be reported to the control node.

#### 3.2 Term Frequency and Inverse Document Frequency (TF-IDF)

Term Frequency (TF)–Inverse Document Frequency (IDF) is a renowned measure for constructing a vector space model in information retrieval. This measure will assess the significance of a word within a document. The significance will increase in proportion to the number of times that a word will appear in a document compared to the inverse proportion of the same word in the whole document collection. In a broader aspect, the TF–IDF measure, which is associated with a term *t*, will take on [19]:

In more formal terms, let  $D = \{d_1, d_2, \dots, d_n\}$  indicate a comprehensive document collection, and *t* indicates a term within the collection. Evaluation of the TF-IDF measure will be according to the below Eq. (2):

$$tf-idf(t, d, D) = tf(t, d) * idf(t, D) \quad (2)$$

To be specific,  $tf(t, d)$  will indicate the term  $t$ 's frequency in document  $d$ , which can be expressed as Eq. (3):

$$tf(t, d) = \frac{f(t, d)}{|d|} \quad (3)$$

In this equation,  $f(t, d)$  will indicate the number of times the term  $t$  will appear in document  $d$ . The denominator will indicate the  $d$ 's dimension, expressed as its own terms' cardinality. Eq. (4) will give the following description of  $idf(t, D)$ :

$$idf(t, D) = \log \frac{|D|}{|\{d|t \in d\}|} \quad (4)$$

Here, the denominator will indicate the number of documents in which there is the occurrence of term  $t$ .

### 3.3 Particle Swarm Optimization (PSO) Algorithm Based Feature Selection

The PSO is made up of a particle population in which every particle indicates a potential solution. After the population's random initialization, every particle will search across the multi-dimensional search space with a particular velocity and update its velocity and position based on two distinct factors: its optimum location so far and the best optimum amongst the whole population [20]. Suppose that  $D$  will indicate a search space's dimension,  $x_{id}(t)$  will indicate the position of the  $i^{\text{th}}$  particle at the  $d^{\text{th}}$  dimension, and  $v_i(t)$  will indicate the velocity of the  $i^{\text{th}}$  particle.  $x_i^{\text{best}}$  will indicate the best previously visited position (up to time  $t$ ) of the  $i^{\text{th}}$  particle and  $x_g^{\text{best}}$  will indicate the global best position of a swarm. In addition,  $c_1, c_2, r_1, r_2$  it will indicate fixed random numbers for learning the procedure. Then, an update of the particle's velocity will be as per the below Eq. (5):

$$v_{id}(t+1) = v_{id}(t) + c_1 r_1 (x_i^{\text{best}} - x_{id}(t)) + c_2 r_2 (x_g^{\text{best}} - x_{id}(t)) \quad (5)$$

Also, update of each object's position will be as per the below Eq. (6):

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (6)$$

In the proposed method, every position will be a vector with the size of the number of features, wherein 0 or 1 will respectively indicate the feature's absence or presence in the vector's  $i^{\text{th}}$  element. Changes in the particle velocity are interpreted as changes in the probability of detecting the particle in a single state. The total number of features obtained from TF-IDF is 242. The particles are encoded randomly during the initialization. During encoding, random features from the 242 features are considered as a subset. On iterating, the optimal subset is obtained. With PSO-based feature selection, 202 features are selected.

### 3.4 Proposed River Formation Dynamics (RFD) Algorithm Based Feature Selection

The following is a summary of the RFD-based feature selection method. There will be assignation of a soil amount to every node. In turn, during the drops' motion, they will either erode their paths or deposit the carried sediment, thereby increasing the nodes' altitudes. The probability of picking the next node will depend upon the decreasing gradient, which is directly proportional to the height difference between the node, at which the droplet resides, and its neighbor. At first, the created environment has a flat gradient; that is, the nodes' altitudes are equal, excluding the goal node retaining a zero height throughout the whole procedure. Then, there is the placement of drops in the initial node for facilitating further site exploration, and thus, lead to the optimal path's detection. At every step, a group of drops will sequentially move along the space and then will carry out erosion on the visited nodes [21].

The first step will involve initializing the algorithm's nodes (*initialize nodes()*) through the definition of a generated set using the site's cell decomposition. Moreover, every node will possess information on whether it has an obstacle as well as extra data, as an example, to determine the required time for traversal and its distance towards the goal. On the other hand, when there is the initialization of the drops (*initializeDrops()*), there will be the placement of a suitable number of drops in the initial node.

The algorithm's execution will be until arrival at the final condition (*endConditionMet()*). This will signify that all the drops have been proceeding along the same path. In addition, for mitigating the computational time, there is the introduction of an upper limit on the number of iterations and a condition for authenticating whether there has been no improvement in the solution by the previous  $n$  loops.

Drops will shift one at a time till either they arrive at the goal or cannot move any further. At this point, the drops will undergo evaporation and will restart in a new loop. The below Eq. (7) will express the probability that the drop  $k$ , which resides in the node  $i$ , will pick the next node  $j$  as follows:

$$P_k(i, j) = \begin{cases} \frac{gradient(i, j)}{sum.(d_j)^\alpha}, & \text{for } j \in V_k(i) \\ \frac{\omega/|gradient(i, j)|}{sum.(d_j)^\alpha}, & \text{for } j \in U_k(i) \\ \frac{\delta}{sum.(d_j)^\alpha}, & \text{for } j \in F_k(i) \end{cases} \quad (7)$$

In this equation,  $V_k(i)$  will indicate the neighbor set has a positive gradient (i.e., the altitude of node  $i$  is higher than that of node  $j$ ),  $U_k(i)$  will indicate the neighbor set was having a negative gradient (i.e., the altitude of node  $j$  is higher than that of node  $i$ ), and  $F_k(i)$  will indicate neighbors having a flat gradient (i.e., similar altitude as node  $i$ ). There will be a selection of neighbors from all the surrounding nodes (up to 8 nodes), excluding those situated within cells that contain an obstacle. While the *gradient's* definition will indicate the altitude difference between consecutive nodes, the coefficients  $\omega$  and  $\delta$  will indicate specific fixed small values. In this case, the *sum* will indicate the sum of the weights of every neighbor from various collections (the numerators). Coefficient  $d_j$  will indicate the length from node  $j$  to the goal, and  $\alpha$  will indicate a convergence tuning coefficient.

The RFD will assume the probability of traversing the edges with an increasing slope. Suppose there is a selection of an edge having an increasing slope. In that case, there will be the performance of a drawing having a low probability (i.e., with every iteration, there will be a decrease). If there is an unsuccessful draw, there will be the drop's evaporation, and the amount of sediment left behind will be equivalent to the amount of sediment by the drop with reduction by a specific parameter.

The best solution's detection will constitute the path analysis (*analyzePaths()*). This process will find the best drop for carrying out the extra erosion.

There is the execution of erosion on all the traversed paths by reducing the nodes' altitudes in direct proportion to the gradient having the successive node. A parameter is used for multiplying the best drop's amount of reduction for enhancing the convergence. Eq. (8) will give the node erosion's formula as below:

$$erosion(i, j) = \begin{cases} \frac{\varepsilon V.gradient(i, j)}{(N - 1).M.pathLength_k}, & \text{for } j \in V_k(i) \\ \frac{\varepsilon U}{|gradient(i, j)|.(N - 1).M.pathLength_k}, & \text{for } j \in U_k(i) \\ \frac{\varepsilon F}{(N - 1).M.pathLength_k}, & \text{for } j \in F_k(i) \end{cases} \quad (8)$$

Here,  $\varepsilon V$  will indicate the parameter connected to the neighbor group with positive gradient,  $\varepsilon U$  will indicate the parameter connected to the neighbor group with negative gradient,  $\varepsilon F$  will indicate the parameter connected to the neighbor group with flat gradient,  $PathLength_k$  will indicate the drop's traverse path length,  $N$  will indicate the number of nodes, and  $M$  will indicate the number of drops.

The final step will involve the addition of a specific small sediment amount to all the nodes for the avoidance of a situation in which all the altitudes are near to 0, which in turn, would result in the gradients being negligible as well ruining all the formed paths. With every loop of the algorithm, there will be a decrease in this quantity.

Feature selection will represent all paths in the RFD as a string, wherein the string's length will be equivalent to the dataset's number of features. In the binary version, '1' will represent the corresponding feature's selection, while '0' will represent the corresponding feature's deselection. The total number of features obtained from TF-IDF is 242. The drops are encoded randomly during the initialization. During encoding, random features from the 242 features are considered as a subset. On iterating, the optimal subset is obtained. With RFD-based feature selection, 198 features are selected.

#### 4 Results and Discussion

Assessment of the algorithm and its quality will be done based on the MovieLens, a historical dataset for the movie recommender systems. This dataset is made up of 100,000,029 anonymous ratings, which are gathered from approximately 6,040 users from 3,952 movies. The primary use of the MovieLens datasets is for assessing a collaborative recommender system for the movie domain. The parameters set for PSO are population size 50, Maximum number of iterations 150, and  $c_1 = c_2 = 2$ . The parameters of RFD are the number of drops 50 and the maximum number of iterations 200.

The user profiles are formatted in MovieLens as in Eq. (9).

$$profile(u) = \{ \langle m_j, v_j \rangle \mid v_j = 1 \text{ if } u \text{ likes } m_j, \\ v_j = -1 \text{ otherwise} \} \quad (9)$$

A MovieLens user  $u$  will express a rate on the movie  $m_j$ . The ratings are given in 5-star format; thus the rating scale  $\in [1; : : : 5]$ , wherein a vote of 1 means an Awful movie, and a vote of 5 is a Must-See movie. To map a five-star rating to that of a binary one, it will be calculated for every user  $u$  within the dataset, and their average rating  $\hat{r}(u)$  will be created in the form of a user profile as shown in (10):

$$profile(u) = \{ \langle m_j, v_j \rangle \mid v_j = 1 \text{ if } r(u, m_j) \geq \hat{r}(u), \\ v_j = -1 \text{ otherwise} \} \quad (10)$$

This section will have discussions about the utilization of the PSO feature selection-without Frequent Pattern Mining (FPM), the PSO feature selection-without FPM + CF, the PSO feature selection-with Systolic Tree FPM, the PSO feature selection-with Systolic Tree FPM + CF, the RFD feature selection-without FPM, the RFD feature selection-without FPM + CF, the RFD feature selection-with Systolic Tree FPM, and the RFD feature selection-with Systolic Tree FPM + CF. Performance of the experiments was done with top  $N=2$  to 18 recommended items. Illustrations of the precision, as well as recall outcomes, have been given in Tabs. 1 and 2, Figs. 1 and 2.

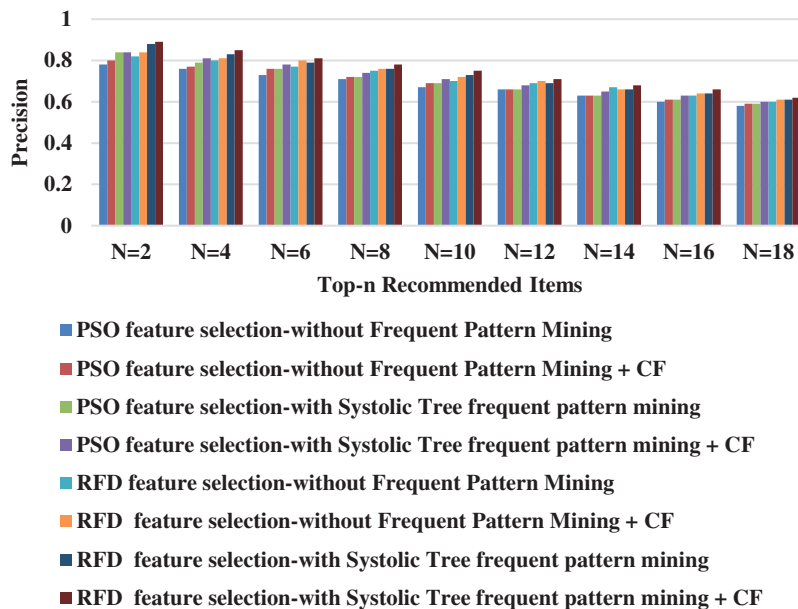


**Table 1:** Precision for RFD feature selection-with systolic tree frequent pattern mining + CF

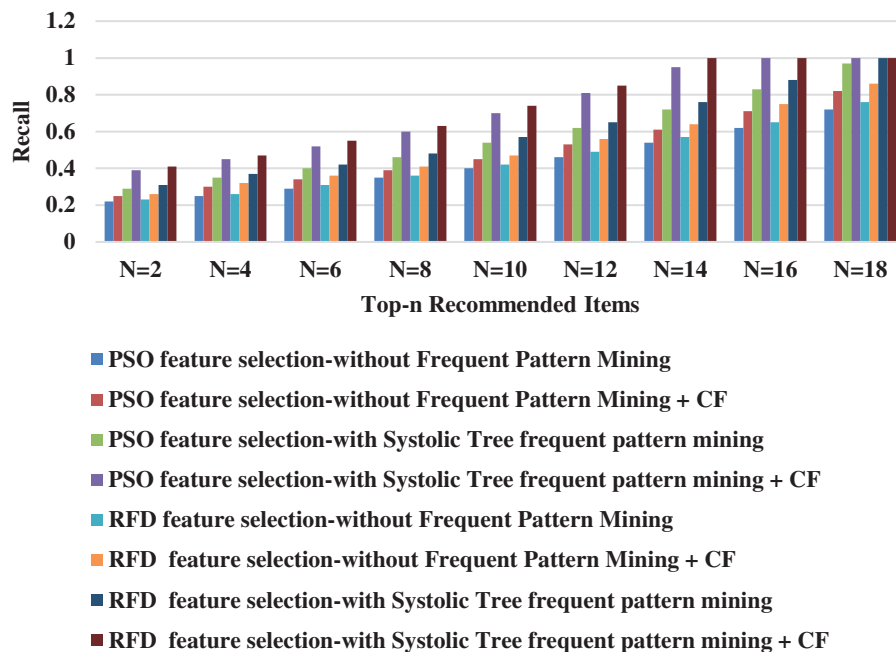
Top-N recommended items	PSO feature selection-without frequent pattern mining	PSO feature selection-without frequent pattern mining + CF	PSO feature selection-with systolic tree frequent pattern mining	PSO feature selection-with systolic tree frequent pattern mining + CF	RFD feature selection-without frequent pattern mining	RFD feature selection-without frequent pattern mining + CF	RFD feature selection-with systolic tree frequent pattern mining	RFD feature selection-with systolic tree frequent pattern mining + CF
N = 2	0.78	0.8	0.84	0.84	0.82	0.84	0.88	0.89
N = 4	0.76	0.77	0.79	0.81	0.8	0.81	0.83	0.85
N = 6	0.73	0.76	0.76	0.78	0.77	0.8	0.79	0.81
N = 8	0.71	0.72	0.72	0.74	0.75	0.76	0.76	0.78
N = 10	0.67	0.69	0.69	0.71	0.7	0.72	0.73	0.75
N = 12	0.66	0.66	0.66	0.68	0.69	0.7	0.69	0.71
N = 14	0.63	0.63	0.63	0.65	0.67	0.66	0.66	0.68
N = 16	0.6	0.61	0.61	0.63	0.63	0.64	0.64	0.66
N = 18	0.58	0.59	0.59	0.6	0.6	0.61	0.61	0.62

**Table 2:** Recall for RFD feature selection-with systolic tree frequent pattern mining + CF

Top-N recommended items	PSO feature selection-without frequent pattern mining	PSO feature selection-without frequent pattern mining + CF	PSO feature selection-with systolic tree frequent pattern mining	PSO feature selection-with systolic tree frequent pattern mining + CF	RFD feature selection-without frequent pattern mining	RFD feature selection-without frequent pattern mining + CF	RFD feature selection-with systolic tree frequent pattern mining	RFD feature selection-with systolic tree frequent pattern mining + CF
N = 2	0.22	0.25	0.29	0.39	0.23	0.26	0.31	0.41
N = 4	0.25	0.3	0.35	0.45	0.26	0.32	0.37	0.47
N = 6	0.29	0.34	0.4	0.52	0.31	0.36	0.42	0.55
N = 8	0.35	0.39	0.46	0.6	0.36	0.41	0.48	0.63
N = 10	0.4	0.45	0.54	0.7	0.42	0.47	0.57	0.74
N = 12	0.46	0.53	0.62	0.81	0.49	0.56	0.65	0.85
N = 14	0.54	0.61	0.72	0.95	0.57	0.64	0.76	1
N = 16	0.62	0.71	0.83	1	0.65	0.75	0.88	1
N = 18	0.72	0.82	0.97	1	0.76	0.86	1	1



**Figure 1:** Precision for RFD feature selection-with systolic tree frequent pattern mining + CF



**Figure 2:** Recall for RFD feature selection-with systolic tree frequent pattern mining + CF

From Fig. 1, it can be seen that the RFD feature selection-with Systolic Tree frequent pattern mining + CF has higher average precision by 9.79% for PSO feature selection-without Frequent Pattern Mining, by 8.01% for PSO feature selection-without Frequent Pattern Mining + CF, by 7.05% for PSO feature selection-with Systolic Tree frequent pattern mining, by 4.7% for PSO feature selection-with Systolic Tree frequent pattern mining + CF, by 4.85% for RFD feature selection-without Frequent Pattern Mining, by 3.16% for RFD feature selection-without Frequent Pattern Mining + CF and by 2.39% for RFD

feature selection-with Systolic Tree frequent pattern mining when compared with various top-N recommended items respectively.

From Fig. 2, it can be seen that the RFD feature selection-with Systolic Tree frequent pattern mining + CF has a higher average recall by 53.33% for PSO feature selection-without Frequent Pattern Mining, by 40.72% for PSO feature selection-without Frequent Pattern Mining + CF, by 24.85% for PSO feature selection-with Systolic Tree frequent pattern mining, by 3.52% for PSO feature selection-with Systolic Tree frequent pattern mining + CF, by 48.59% for RFD feature selection-without Frequent Pattern Mining, by 35.81% for RFD feature selection-without Frequent Pattern Mining + CF and by 20.02% for RFD feature selection-with Systolic Tree frequent pattern mining when compared with various top-N recommended items respectively. Though the precision and recall are satisfactory, the cold start and sparsity problems are not addressed.

## 5 Conclusion

Over the past years, recommender systems have increasingly gained wide acceptance and thus, have leveled the ground for new sale opportunities in e-commerce. This work has devised a CF method for offering movie recommendations to e-commerce customers. On the Internet, the CF is a renowned method for offering recommendations. As a subfield of the data mining techniques, sequential pattern mining will deal with discovering such patterns inside the sequence data. In feature selection, the total search space for the detection of the most appropriate and non-redundant features includes all the potential subsets. This work has presented a novel technique of feature selection technique that will pick the most informative features while omitting the irrelevant or redundant ones. Embedding of this technique is done in the PSO and the RFD. For the omission of irrelevant or redundant features, it is essential to figure out the relationship between diverse features. Being a heuristic algorithm, the PSO can minimize the complexity as well as acquire a near-optimal solution. The RFD is a heuristic optimization algorithm based on replicating how water drops will form rivers via ground erosion and sediment deposition. It is evident from the experimental outcomes that, when compared against several top-N recommended items, the RFD feature selection-with Systolic Tree frequent pattern mining + CF has a higher average precision by 9.79% for the PSO feature selection-without Frequent Pattern Mining, by 8.01% for the PSO feature selection-without Frequent Pattern Mining + CF, by 7.05% for the PSO feature selection-with Systolic Tree frequent pattern mining, by 4.7% for the PSO feature selection-with Systolic Tree frequent pattern mining + CF, by 4.85% for the RFD feature selection-without Frequent Pattern Mining, by 3.16% for the RFD feature selection-without Frequent Pattern Mining + CF, and by 2.39% for the RFD feature selection-with Systolic Tree frequent pattern mining. This work is the initial investigation that forms the basis for exploring hybrid algorithms to improve the metaheuristic algorithms effectiveness for recommendations. Further work should investigate the efficacy of the proposed methods with a real-time dataset.

**Acknowledgement:** Thanks to the authors and reviewers.

**Funding Statement:** The authors received no specific funding for this study.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] M. Ghazanfar and A. Prugel-Bennett, "An improved switching hybrid recommender system using naive Bayes classifier and collaborative filtering," in *The 2010 IAENG Int. Conf. on Data Mining and Applications*, Hong Kong, pp. 17–19, Mar 2010.

- [2] Q. Zhang, G. Zhang, J. Lu and D. Wu, "A framework of a hybrid recommender system for personalized clinical prescription," in *Int. Conf. on Intelligent Systems and Knowledge Engineering (ISKE)*, Dalian, China, IEEE, pp. 189–195, November 2015.
- [3] Y. Afoudi, M. Lazaar and M. Al Achhab, "Impact of feature selection on content-based recommendation system," in *2019 Int. Conf. on Wireless Technologies, Embedded and Intelligent Systems (WITS)*, MOROCCO, pp. 1–6, 2019.
- [4] U. Kuzelewska, "Clustering algorithms in hybrid recommender system on movielens data," *Studies in Logic, Grammar and Rhetoric*, vol. 37, no. 1, pp. 125–139, 2018.
- [5] R. Trivonanda, R. Mahendra, I. Budi and R. A. Hidayat, "Sequential pattern mining for e-commerce recommender system," in *2020 Int. Conf. on Advanced Computer Science and Information Systems (ICACSIS)*, Indonesia, pp. 393–398, 2020.
- [6] M. Kolahkaj and M. Khalilian, "A recommender system by using a classification based on frequent pattern mining and J48 algorithm," in *2015 2nd Int. Conf. on Knowledge-Based Engineering and Innovation (KBEI)*, Iran, pp. 780–786, 2015.
- [7] J. J. Junior, F. G. Vilasbôas and L. N. de Castro, "The influence of feature selection on job clustering for an e-recruitment recommender system," in *Int. Conf. on Artificial Intelligence and Soft Computing*, Springer, Cham, vol. 5, pp. 176–187, 2020.
- [8] M. Rostami, K. Berahmand, E. Nasiri and S. Forouzande, "Review of swarm intelligence-based feature selection methods," *Engineering Applications of Artificial Intelligence*, vol. 100, pp. 1–33, 2021.
- [9] J. Bobadilla, A. González-Prieto, F. Ortega and R. Lara-Cabrera, "Deep learning feature selection to unhide demographic recommender systems factors," *Neural Computing and Applications*, vol. 10, pp. 1–18, 2020.
- [10] M. Fang, N. Z. Gong and J. Liu, "Influence function-based data poisoning attacks to top-n recommender systems," in *Proc. of the Web Conf. 2020*, Taiwan, vol. 5, pp. 3019–3025, 2020.
- [11] A. Gogna and A. Majumdar, "DiABIO: Optimization-based design for improving diversity in recommender system," *Information Sciences*, vol. 378, pp. 59–74, 2017.
- [12] S. Bansal and N. Baliyan, "Bi-MARS: A bi-clustering based memetic algorithm for recommender systems," *Applied Soft Computing*, vol. 97, pp. 1067–1085, 2020.
- [13] M. Hamada and M. Hassan, "Artificial neural networks and particle swarm optimization algorithms for preference prediction in multi-criteria recommender systems," *Informatics Multidisciplinary Digital Publishing Institute*, vol. 5, no. 2, pp. 25–32, 2018.
- [14] T. Mohammadpour, A. M. Bidgoli, R. Enayatifar and H. H. S. Javadi, "Efficient clustering in collaborative filtering recommender system: A hybrid method based on genetic algorithm and gravitational emulation local search algorithm," *Genomics*, vol. 111, no. 6, pp. 1902–1912, 2020.
- [15] R. Katarya and O. P. Verma, "A collaborative recommender system enhanced with particle swarm optimization technique," *Multimedia Tools and Applications*, vol. 75, no. 15, pp. 9225–9239, 2016.
- [16] L. Shen, Q. Liu, G. Chen and S. Ji, "Text-based price recommendation system for online rental houses," *Big Data Mining and Analytics*, vol. 3, no. 2, pp. 143–152, 2020.
- [17] P. Nitu, J. Coelho and P. Madiraju, "Improvising personalized travel recommendation system with recency effects," *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 139–154, 2021.
- [18] D. Sugumar and P. L. Bose, "Discovering probabilistic frequent sequential patterns in uncertain databases under systolic tree," *International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)*, vol. 3, no. 2, pp. 201–205, 2020.
- [19] U. Erra, S. Senatore, F. Minnella and G. Caggianese, "Approximate TF-IDF based on topic extraction from massive message stream using the GPU," *Information Sciences*, vol. 292, pp. 143–161, 2015.
- [20] Z. Shojaee, S. A. Shahzadeh Fazeli, E. Abbasi and F. Adibnia, "Feature selection based on particle swarm optimization and mutual information," *Journal of AI and Data Mining*, vol. 9, no. 1, pp. 39–44, 2021.
- [21] G. Redlarski, A. Pałkowski and M. Dąbkowski, "Using river formation dynamics algorithm in mobile robot navigation," *Solid State Phenomena Trans Tech Publications Ltd.*, vol. 198, pp. 138–143, 2013.