mathematics

Article

Design and Analysis of a Cluster-Based Intelligent Hybrid Recommendation System for E-Learning Applications

Sundaresan Bhaskaran, Raja Marappan * and Balachandran Santhi

School of Computing, SASTRA Deemed University, Thanjavur 613401, India; bhaskaran@mca.sastra.edu (S.B.); shanthi@cse.sastra.edu (B.S.)
* Correspondence: raja_csmath@cse.sastra.edu

Abstract: Recently, different recommendation techniques in e-learning have been designed that are helpful to both the learners and the educators in a wide variety of e-learning systems. Customized learning, which requires e-learning systems designed based on educational experience that suit the interests, goals, abilities, and willingness of both the learners and the educators, is required in some situations. In this research, we develop an intelligent recommender using split and conquer strategy-based clustering that can adapt automatically to the requirements, interests, and levels of knowledge of the learners. The recommender analyzes and learns the styles and characteristics of learners automatically. The different styles of learning are processed through the split and conquer strategy-based clustering. The proposed cluster-based linear pattern mining algorithm is applied to extract the functional patterns of the learners. Then, the system provides intelligent recommendations by evaluating the ratings of frequent sequences. Experiments were conducted on different groups of learners and datasets, and the proposed model suggested essential learning activities to learners based on their style of learning, interest classification, and talent features. It was experimentally found that the proposed cluster-based recommender improves the recommendation performance by resulting in more lessons completed when compared to learners present in the no-recommender cluster category. It was found that more than 65% of the learners considered all criteria to evaluate the proposed recommender. The simulation of the proposed recommender showed that for learner size values of <1000, better metric values were produced. When the learner size exceeded 1000, significant differences were obtained in the evaluated metrics. The significant differences were analyzed in terms of a computational structure depending on $|L|$, the recommendation list size, and the attributes of learners. The learners were also satisfied with the accuracy and speed of the recommender. For the sample dataset considered, a significant difference was observed in the standard deviation σ and mean μ of parameters, in terms of the *Recall* (*List*, *User*) and *Ranking Score* (*User*) measures, compared to other methods. The devised method performed well concerning all the considered metrics when compared to other methods. The simulation results signify that this recommender minimized the mean absolute error metric for the different clusters in comparison with some well-known methods.

Keywords: e-learning; intelligent optimization; personalization; recommendation system; hybrid recommender; cluster-based recommender

Citation: Bhaskaran, S.; Marappan, R.; Santhi, B. Design and Analysis of a Cluster-Based Intelligent Hybrid Recommendation System for E-Learning Applications. *Mathematics* **2021**, *9*, 197. https://doi.org/10.3390/math9020197

Received: 21 December 2020
Accepted: 11 January 2021
Published: 19 January 2021

1. Introduction

Currently, e-learning has replaced conventional learning systems to ensure that better objectives are achieved by all learners [1–4]. The usage of web enhancements to provide online or offline learning to students at any time is called e-learning, which is considered more effective than conventional learning techniques. Personalized e-learning allows learners to access resources wherever and whenever needed in a useful manner [5–7].

The current research in e-learning focuses on the development of recommendation methodologies that are expected to achieve better performance compared to the existing recommendation strategies. Hence, it is important to develop a better recommendation system to provide better service to learners. The recommendation system proposed herein consists of some important subsystems, namely, the Learner Subsystem, Domain Subsystem, Application Subsystem, Adaptation Subsystem, and Session Subsystem. This research presents the design of some strategies needed to provide better recommendations compared to the existing state-of-the-art techniques. The proposed method was executed on an educational dataset with 1000 learners. The experimental results allowed us to conclude that the learners from the simulation cluster could complete a course with reduced computational time and more lessons when compared to the no-recommender cluster. It was also found that the proposed model intelligently recommends learning resources based on the characteristics and styles of learning.

Some research gaps in well-known recommendation systems involve greater differences in the measures of performance and greater computational complexity required, resulting in lower recommendation accuracy. Hence, it is important to develop new recommendation methodologies to offset these problems in well-known strategies in order to find solutions to different real-world issues. To address the above-mentioned drawbacks, in this research we describe the development of an intelligent recommender that can adapt automatically to the requirements, interests, and levels of knowledge of the learners. The recommender automatically analyzes and learns the styles and characteristics of learners. The different styles of learning are processed through split and conquer strategy-based clustering. Then, the system provides intelligent recommendations by evaluating the ratings of frequent sequences.

The proposed intelligent hybrid recommender system applies the following new strategies:

- The datasets are separated into different clusters using the split and conquer strategy;
- Better recommendations are updated by generating recommendations from each cluster;
- A cluster-based linear pattern mining strategy is applied to identify the maximal large cluster sequences;
- The maximal large sequences are enhanced using the linear pattern pruning strategy;
- The styles of learning are identified based on the characteristics of learners; and
- The preferences of learning styles are evaluated in different dimensions.

The definitions and notation applied in this recommender model are presented in Section 2. A literature survey of the existing recommendation methods and the need for new recommendation strategies are discussed in Section 3, and the proposed method is explained in Section 4. The experimental results and an analysis of the proposed method are presented in Section 5, and Section 6 concludes this research.

2. A Literature Survey on Recommendation Systems and the Need for New Recommendation Strategies

Some existing e-learning systems were designed based on the methods of learning, types of information, learner characteristics, and other specific features of operational procedures [8–11]. Content-based collaborative filtering (CF) and mining procedures were developed [12], as were rule-based customized learning systems with fuzzy theory [13], a mutual filtering procedure with maximum likelihood estimation [14], a mutual filtering recommendation for small datasets [15], a Bayesian model-based learning system [16], and a domain perspective model-based recommendation framework [17].

A semantic recommendation with an ontology-based approach was proposed [18], and a self-organizing map-based e-learning recommendation was developed using artificial neural networks [19]. A customized recommendation model using machine learning and clustering was developed to analyze the learning paths of learners [20]. Customized object-based learning with different styles of learning was analyzed using a clustering algorithm [21], and the learning styles and knowledge levels of learners were evaluated using a content filtering approach [22].

The recently developed recommender systems can be classified into different groups, as outlined below [23–73].

2.1. User Profile Recommendation Systems

Personalized recommendation systems have been developed recently for some domains. The User Profile Oriented Diffusion (UPOD) strategy was developed to analyze learner profiles [74]. This methodology makes specific recommendations based on different operations that are executed in the training stage. The referral stage generates recommendations based on learner profile features.

2.2. Content Recommendation Systems

The convolutional neural network (CNN) method with content-based recommendation was discussed in [75]. This method is used to obtain the concealed factors in different media applications. In this method, the textual information is processed in recommending the required contents.

2.3. Hybrid Recommendation Systems

Some hybrid strategies have been developed for recommending movie-based applications [76]. Content-based filtering was also applied in recommending better movies to users. Personalized learning with a hybrid strategy was developed in [77]. The recommendation system gives personalized recommendations using visualizations.

2.4. Filter-Based Recommendation Systems

Some CF-based recommendations have been implemented for user travel recommendations [78]. This recommendation scheme obtained good performance when compared to other schemes.

2.5. Feature-Based Recommendation Systems

A feature-based recommendation system to solve the ERP System and E-Agribusiness datasets was developed [79], enhanced CF was developed for solving *MovieLens* applications [80], a group recommender strategy was developed in [81,82], and similarity-based recommender systems for different applications were developed in [83–103].

Critiques of the well-known recommender systems in e-learning [23–30] include the difference in the average absolute error, lower accuracy in the recommendation, and longer computing time required during the recommendation [33–45]. To address these problems in the existing well-known methods, the recommender proposed herein develops an intelligent recommender that can adapt automatically to the requirements, interests, and levels of knowledge of the learners. The recommender automatically analyzes and learns the styles and characteristics of learners. The different styles of learning are processed through a split and conquer based clustering strategy. Then, the system provides intelligent recommendations by evaluating the ratings of frequent sequences.

3. Notation and Definitions

The following notation and definitions are used in the proposed model [46–72,103].

1. Learning Items
   The learning items of the resources are defined as the collection of learning objects and are defined as *Learning-Items* = $\{Item_1, Item_2, Item_3, \ldots, Item_n\}$.
2. Sequence of Item Sets
   The sequence of item sets defines the ordered sequence: $(Item_1, Item_2, Item_3, \ldots, Item_n)$.
3. Subset of a Sequence
   A sequence $(x_1, x_2, x_3, \ldots, x_n)$ is a subset of another sequence $(y_1, y_2, y_3, \ldots, y_n)$ if $x_1 \subseteq y_{a1}, x_2 \subseteq y_{a2}, x_3 \subseteq y_{a3}, \ldots, x_n \subseteq y_{an}$ where $a1, a2, a3, \ldots, a_n$ are integers such that $a1 < a2 < a3 < \ldots < a_n$.

4. Maximal sequence
   If a sequence x is not a subset of some other sequence, then x is said to be a maximal sequence.
5. Linear Pattern
   A linear pattern is defined as a maximal sequence.
6. *Support*(s)
   For a sequence s, *Support*(s) is defined as the proportion of learners supporting s.
7. *Rating* (L)
   The rating of a learner L is defined in vector form as
   Rating(L) = (r_1, r_2, r_3, ... , r_j) where r_1, r_2, r_3, ... , r_j define the degree of learners' knowledge for the specific module used in learning.
8. *Expected Rating* (L)
   If S_L defines a set of frequent sequences, then the expected rating is defined as follows:

$$\overline{r} = \frac{1}{|S_L|} \sum_{j \in S_L} r_j. \tag{1}$$

9. *Similarity* (a,b)
   The similarity measurement for two learners a and b is computed as follows:

$$Similarity(a,b) = \frac{\sum_{j \in S_a \cap S_b} (r_{aj} - \overline{r}_a)\left(r_{bj} - \overline{r}_b\right)}{\sqrt{\sum_{j \in S_a \cap S_b} (r_{aj} - \overline{r}_a)^2 \sum_{j \in S_a \cap S_b} (r_{bj} - \overline{r}_b)^2}}. \tag{2}$$

10. *Prediction* (a,j)
    The prediction estimation for a learner a with sequence j is defined as follows:

$$Prediction(a,j) = \overline{r}_a + w_{aj} \sum_{a \in L_j} Similarity(a,b)\left(r_{bj} - \overline{r}_b\right). \tag{3}$$

11. *Normalized vector*
    The normalized vector w_{aj} is defined as follows:

$$w_{aj} = 1/ \sum_{a' \in L_j} Similarity(a,b). \tag{4}$$

12. *Recall* (*List*, *User*)
    Let $g(L)$ be the item count that is associated with the valid target users, $Valid_{Users}$, in the recommendation list L and testing set. Let I_{User} be the total item count related to the valid user $User \in Valid_{Users}$. Then, *Recall* (*List*, *User*) is defined as follows:

$$Recall(List,\ User) = \frac{g(L)}{I_{User}}. \tag{5}$$

13. *Precision* (*List*, *User*)
    $Precision\ (List,\ User)$ is given as

$$Precision(List,\ User) = \frac{g(L)}{|L|}. \tag{6}$$

14. *Rank* (i)
    The metric *Rank*(i) is represented using the following equation:

$$Rank(i) = \frac{\text{The item position in L}}{\text{The number of items initially unknown to the user}}. \tag{7}$$

15. *Ranking Score* (*User*)

$$Ranking\ Score\ (User) = \sum_{j \in I_{User}} \frac{Rank(j)}{|I_{User}|}. \quad (8)$$

16. Mean Absolute Error (MAE)
For each of the respective appraisals and evaluations of values of p_i and q_i, the MAE is defined as follows:

$$\mathrm{MAE} = \frac{\sum_{i=1}^{N} |p_i - q_i|)}{\mathrm{N}}. \quad (9)$$

4. The General Structure of the Proposed Methodology

An intelligent hybrid recommender is developed in this research that permits both the learners and the educators to use teaching resources effectively. The hybrid intelligent recommender is implemented for all the learning courses and recommends the essential learning resources based on the learner preferences, styles of learning, and characteristics [77].

The proposed intelligent recommender consists of different subsystems: the Domain Subsystem, Learner Subsystem, Application Subsystem, Adaptation Subsystem, and Session Subsystem. The operations of each subsystem are tabulated in Table 1. The overall recommendation system architecture, which comprises these subsystems, is depicted in Figure 1. The recommendation component architecture of the proposed system is shown in Figure 2.

Table 1. Subsystems of the proposed recommender.

Subsystem	*Operations*
Domain Subsystem	Storing the learning resources and different components
Learner Subsystem	Extracting the complete information and features of the learners
Application Subsystem	Applying the operational rules to identify the requirements of learners
Adaptation Subsystem	Identifying the intelligent recommendations to the learners
Session Subsystem	Controlling all subsystems along with the main system

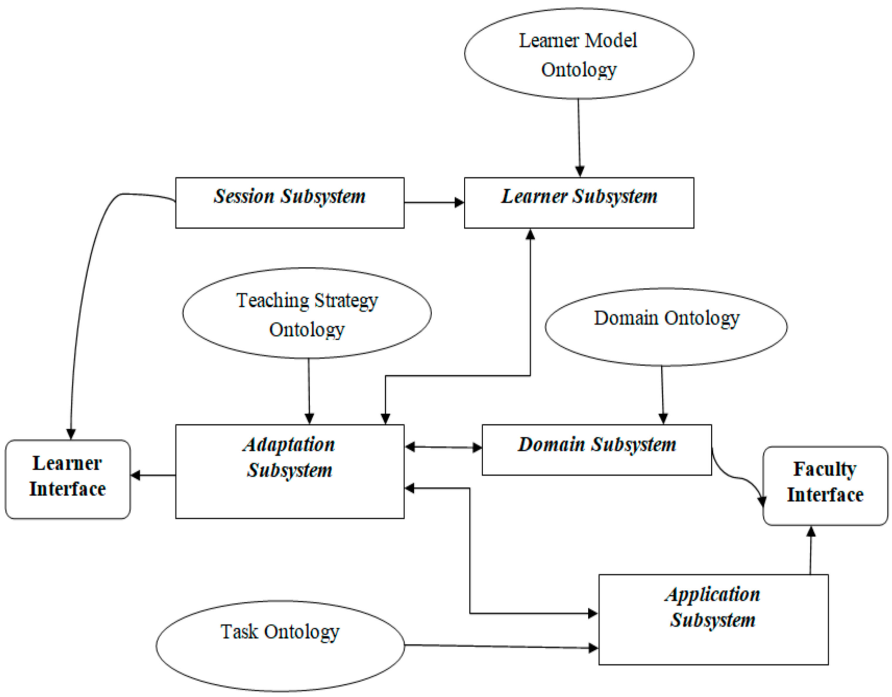

Figure 1. The overall recommendation system architecture.

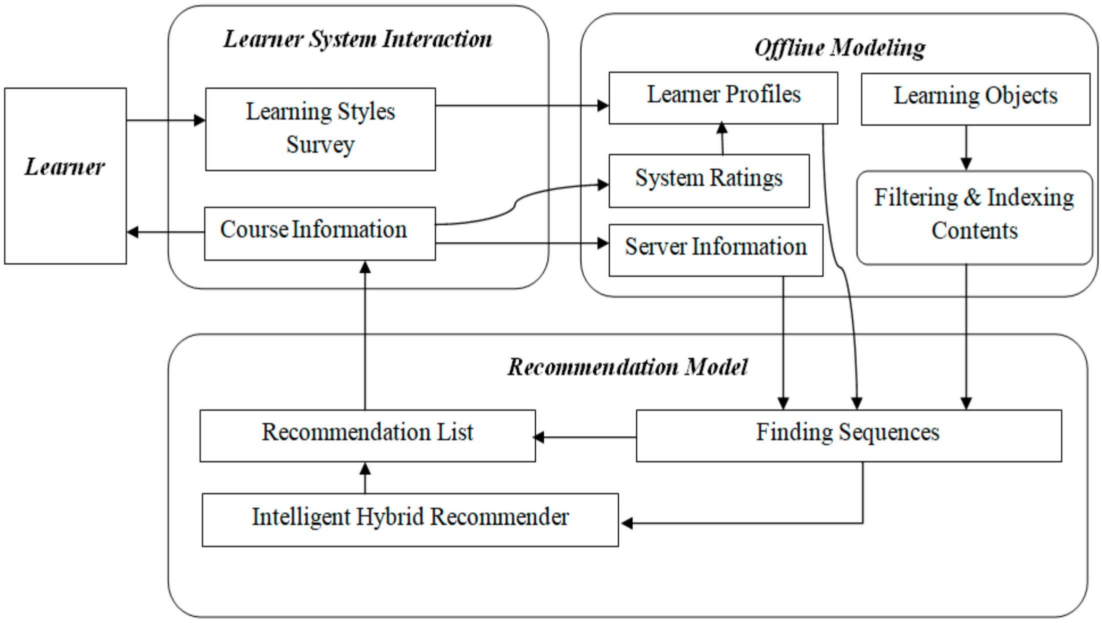

Figure 2. The recommendation component architecture of the proposed system.

4.1. Novelty and Advantages of the Proposed Method

The devised recommender applies the following strategies:

- Splitting the datasets into various clusters via the split and conquer strategy;
- Generating recommendations in each cluster and updating the recommendations;
- A cluster-based linear pattern mining strategy;
- Finding the maximal large sequences using a linear pattern pruning strategy;
- Identifying the styles of learning based on the characteristics of learners; and
- Evaluating the preferences of learning styles in different dimensions.

The following are the advantages of the proposed recommender algorithm:

- Reduced deviation in the performance measurements when compared to well-known methods;
- Reduced computational complexity of the recommendation process;
- Increased accuracy in the recommendation list generation;
- Better recommendation generation from each cluster;
- Identification of the learners' learning styles;
- Analysis of learning style preferences using the Index of Learning Styles strategy; and
- Evaluation of variations in learner preferences across the different dimensions.

4.2. The Proposed Intelligent Hybrid Recommender

The proposed intelligent hybrid recommender model takes the learning resources, *Learning-Resources*, and the learner list, *Learner-List*, as its inputs. The model outputs a better recommendation list to the learners. The overall flowchart of the proposed intelligent hybrid recommender is shown in Figure 3. The general structure of the proposed model is depicted in Algorithm 1. *Learning-Resources* and *Learner-List* are given as inputs to this model. *Learning-Resources* is the contents or materials which are divided into different units, each of which consists of finite lessons. Each lesson consists of different topics such as an introduction, overview, applications, tutorials, tests, and exercises. *Learner-List* represents the faculty and learner information. The faculty prepares different learning components and accesses the appropriate authoring tool. The learners can have different learning characteristics such as requirements, preferences, and methods of learning. The learning style characteristics are identified by evaluating preferences for the learning styles in different dimensions.

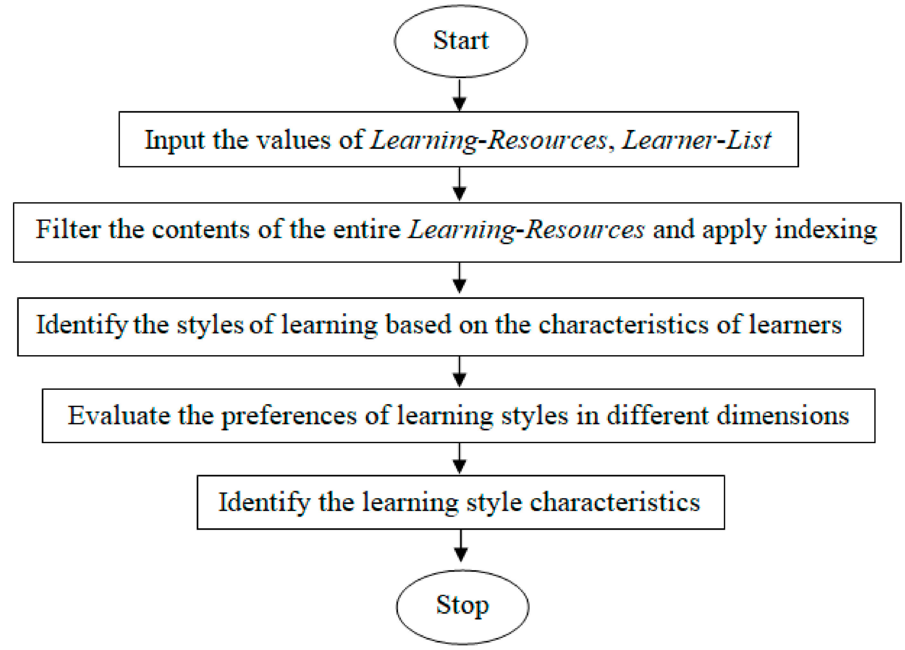

Figure 3. The overall flowchart of the proposed intelligent hybrid recommender.

Algorithm 1 General Structure of the Proposed Intelligent Hybrid Recommender

```
Input: Learning-Resources, Learner-List
Output: Identification of learning style characteristics
Filter the contents of the entire Learning-Resources and apply indexing
Divide Learning-Resources into the required number of different modules and lessons
Introduce the different examples, tests, and activities for each of the lessons and modules
Define the components to add new learning resources
Identify the styles of learning based on the characteristics of learners
Evaluate the preferences for learning styles in different dimensions
Process the information
Identify the perception of information
Identify the reception of information
Understand the information based on different learners
Print the learning style characteristics
```

The learning processes of the learners can be organized by the decisions made by the learners themselves. Depending on the preferences of the learners, they can choose their strategies in generating different learning activities. These learning activities are maintained within the proposed intelligent system. The proposed cluster-based linear pattern mining algorithm is applied to extract the functional patterns of the learners. These patterns can also be used to analyze the complete historical information of the learners, starting from the learning of a specific module and lessons to their successful or unsuccessful completion of the modules and lessons.

The proposed cluster-based linear pattern mining strategy is depicted in Figure 4. The learners are completely clustered and their functional patterns are analyzed as shown in the proposed Algorithm 2. The algorithm starts with the set *Learning-Items* $\{Item_1, Item_2, Item_3, \ldots, Item_n\}$ and the sequence of item sets $(Item_1, Item_2, Item_3, \ldots, Item_n)$. The maximal sequence among all the defined sequences of items is identified, and *Support*(s), the support for sequence s, is evaluated. Then, we obtain the maximal and large sequences of a pattern and perform operations such as sorting, identifying the item sets with maximum size, linear sequence transformation, sequencing operation, and maximal operation. Finally, the maximal large sequences of the cluster are identified so that the functional patterns of the learners can be extracted.

Algorithm 2 Proposed Cluster-Based Linear Pattern Mining

Input: Learning items list, the sequence of items in the cluster
Output: The maximal large sequences of the cluster
1: Define the set *Learning-Items* = {Item1, Item2, Item3, . . . , Item_n}
2: Define the sequence of Item sets: (Item1, Item2, Item3, . . . , Item_n)
3: Identify the maximal sequence among all the defined sequences of item sets
4: Evaluate *Support*(*s*), the support for sequence *s*
5: Obtain the maximal and large sequences of a linear pattern
6: Sort the learner sequences based on the primary key *Learner-Id*
7: Identify the item sets with maximum size
8: Apply transformation of a learner sequence
9: Determine the expected sequences applying the sequencing operation
10: Apply the maximal operation to reduce data redundancy
11: Identify the maximal large sequences of the cluster

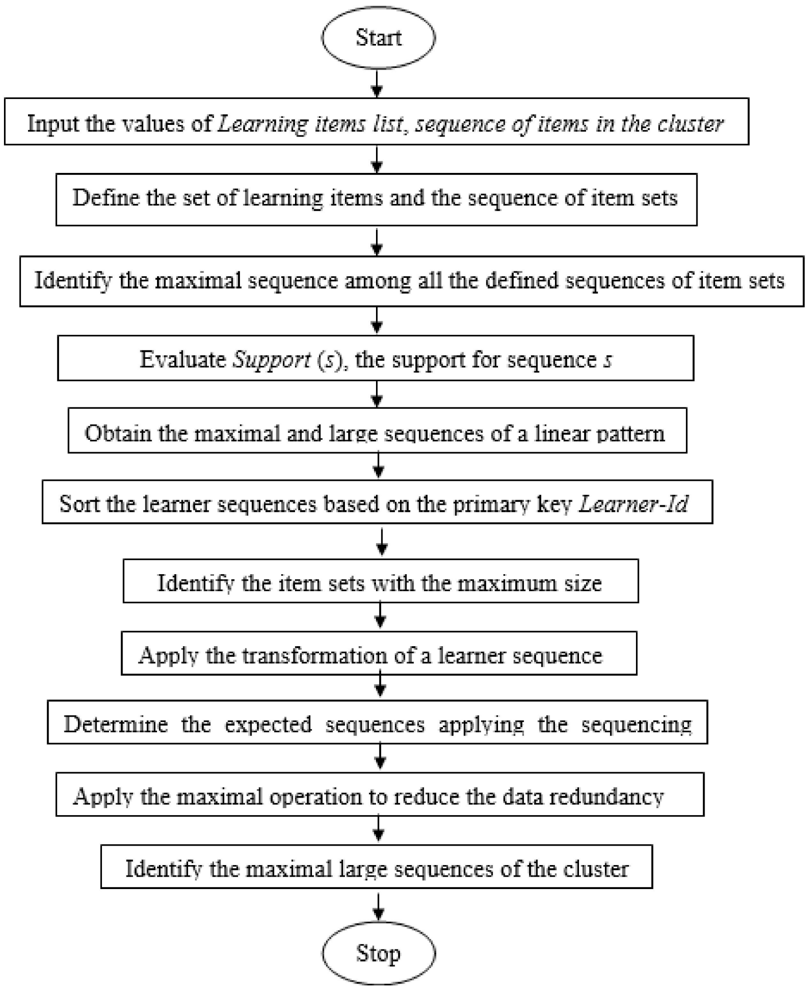

Figure 4. The proposed cluster-based linear pattern mining.

The linear patterns which are the subsets of other linear patterns are examined and pruned to obtain the maximal large sequences; redundant data are thereby reduced [24]. The flowchart of the proposed linear pattern pruning strategy is depicted in Figure 5, and the proposed linear pattern pruning algorithm is shown in Algorithm 3.

Algorithm 3 Proposed Linear Pattern Pruning

Input: Linear patterns
Output: Maximal large sequences
1: Identify the set of sequences
2: For all the identified sequences
3: For all the subsequences in the identified sequences
4: If the subsequence is not in the identified sequences, then
5: Prune the subsequence from the sequence list
6: Print the maximal large sequences

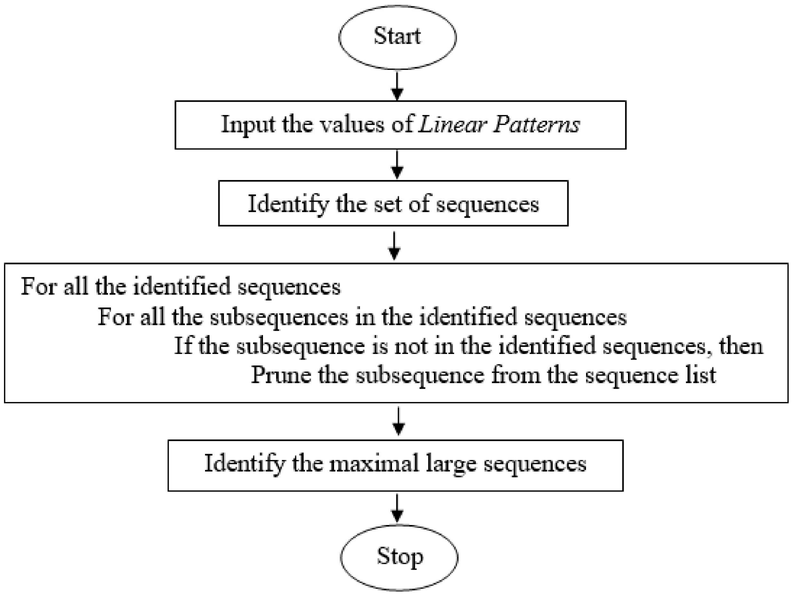

Figure 5. The proposed linear pattern pruning strategy.

Once the learners finish the sequence of learning resources, the proposed cluster-based evaluation evaluates the learners' gained knowledge. The flowchart of the cluster-based evaluation is shown in Figure 6. The corresponding algorithm is sketched in Algorithm 4. This algorithm interprets the results based on the learners' percentages of right answers. The sets of similar learners are also identified.

Algorithm 4 Proposed Cluster-Based Evaluation

Input: Completion of learning resources in the cluster
Output: Evaluation of the learners' gained knowledge in the cluster
1: Check if the sequence of learning resources is completed by the learner
2: Define the levels of learner rating
3: Interpret the results based on the percentage of right answers within the cluster
4: Identify the set of similar learners, if required

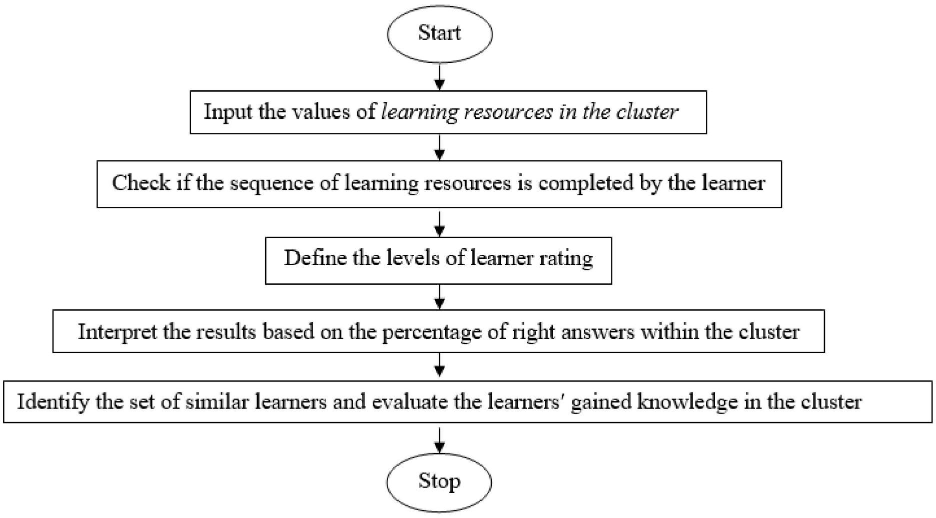

Figure 6. The proposed cluster-based evaluation.

The flowchart of the proposed cluster-based intelligent hybrid recommender is shown in Figure 7, and the algorithm is presented in Algorithm 5. The algorithm splits the input resources into different clusters by applying the divide and conquer strategy and applies the proposed algorithms with the new recommendation and enhanced CF strategies to obtain a better recommendation list for the learners.

Algorithm 5 Proposed Cluster-Based Intelligent Hybrid Recommender

Input: *Learning-Resources*, *Learner-List*
Output: *Recommendation-List*, the better recommendation list for the learners
1: Split *Learning-Resources* and *Learner-List* into the required number of clusters
2: For each cluster, apply the following operations:
3: Apply Algorithm 1 to identify the learning style characteristics of the learners
4: Apply Algorithms 2 and 3 to identify the maximal large sequences of the cluster
5: Apply Algorithm 4 to evaluate the learners' gained knowledge in the cluster
6: Apply the following recommendation strategy:
7: Define the rating vector of the learner using $Rating(L)$
8: Compare the learners' ratings
9: Evaluate the expected rating of the learner using the weighted mean
10: Determine $Similarity(a, b)$ for learners using Equation (2)
11: Compute $Prediction(a, j)$ for a learner a with sequence j
12: Recommend the required activities to the learner based on the evaluation
13: Select the best recommendation from the recommended activities in all clusters evaluation

There are different methods available to explore and analyze the different learning style preferences of learners. The Index of Learning Styles is an objective method used to evaluate the learning style preferences of learners in different dimensions [32,33]. Some of these learning style characteristics are tabulated in Table 2 [32,33].

Table 2. Learning style characteristics of learners.

Active	Reflexive
Tasks in the clusters	Tasks outside of the clusters
Willingness to add new resources	Requiring time to think
Experimental	Theoretical
Visual	Verbal
Images, flowcharts	Complete paragraphs
Global	Sequential
Linear understanding	Fast understanding

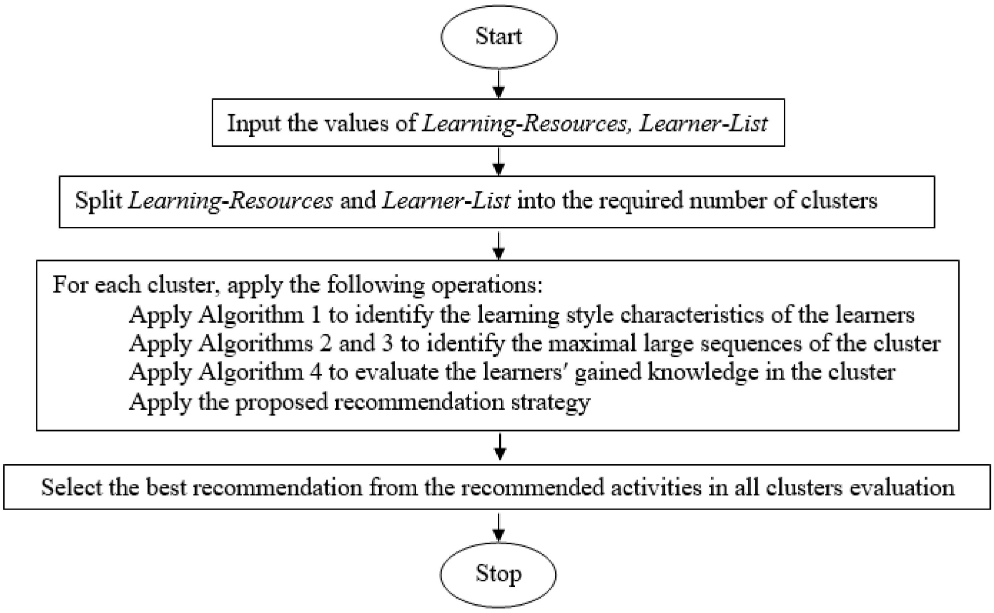

Figure 7. The proposed cluster-based intelligent hybrid recommender.

In the proposed method, the learning style preferences are analyzed using the objective Index of Learning Styles strategy; it evaluates variations in learner preferences across the following different dimensions:

- How to process information for learners;
- Learners' perceptions of information;
- Learners' reception of information; and
- Understanding of information based on different learners.

The sequential pattern mining strategy consists of the following different steps which were applied to a database consisting of the columns Learner-Id, Resource Access Time, and Transaction Access Path.

Sorting: The database was sorted using the column Learner-Id. Access to the learning resources was sorted based on the access times. These results are tabulated in Table 3.

Table 3. Learning resource access based on sorting.

Learner Id	*Resource Access Time*	*Transaction Access Path*
1	1 March 2020	Module 1 Lesson 1 Example, Introduction
1	2 March 2020	Module 1 Lesson 2 Example, Introduction
1	3 March 2020	Module 1 Lesson 3 Introduction
2	4 March 2020	Module 1 Lesson 1 Introduction
2	5 March 2020	Module 1 Lesson 1 Example
2	6 March 2020	Module 1 Lesson 2 Introduction
3	7 March 2020	Module 1 Lesson 2 Applications
3	8 March 2020	Module 1 Lesson 3 Exercises
4	9 March 2020	Module 1 Lesson 1 Example
5	10 March 2020	Module 1 Lesson 1 Introduction

Finding the large item sets: The large item sets were identified. The outcomes of the mapping of large learning item sets are presented in Table 4.

Transformation: This replaces every transaction in a transformed linear sequence with a set of large item sets. The result of the transformation step is shown in Table 5.

Sequencing: In this step, the algorithm applies a set of large item sets to obtain the expected sequences in a fixed number of passes. The large sequences obtained are shown in Table 6.

Table 4. Mapping of large learning item sets.

Large Learning Item Sets	*Mapping*
Module 1 Lesson 1 Introduction	A
Module 1 Lesson 1 Overview	B
Module 1 Lesson 1 Applications	C
Module 1 Lesson 1 Flow diagram	D
Module 1 Lesson 1 Limitations	E
Module 1 Lesson 1 Example 1	F
Module 1 Lesson 1 Example 2	G
Module 1 Lesson 1 Example 3	H
Module 1 Lesson 1 Exercises 1	I
Module 1 Lesson 1 Exercises 2	J
Module 1 Lesson 1 Exercises 3	K
Module 1 Lesson 1 Quiz 1	L

Table 5. Transformation Function based on *Learner-Id*.

Learner-Id	**Transformation Function**
1	<(AB) (CD) (EFGH) (IJ) (KL)
2	<(ABCD) (EFG) (HIJ) (KL)
3	<ABCD) (FGH) (IJKL)
4	<(CDEF) (GH) (IJKL)>
5	<(CDFGH) (IJKL)>

Table 6. Large 3, 4, and 5 sequences.

Size	**Sequence**	**Support**
3	EFG	1
3	HIJ	1
3	FGH	1
4	EFGH	1
4	ABCD	1
4	IJKL	2
5	CDFGH	1

Pruning: The sequential patterns contained within other sequential patterns were pruned to reduce the information redundancy.

Cluster generation based on the learning style characteristics is shown in Table 7. Based on the questionnaires, 16 clusters were constructed to determine the profile of the simulation learners' group. The clusters were constructed for different combinations of learning styles.

Table 7. Cluster generation based on learning style characteristics

Cluster Number	**Learning Styles**	**Number of Learners**
1	active, sensing, sequential, global	26
2	reflexive, intuitive, sensing, global	28
3	visual, global, verbal, sensing	42
...	...	...
...	...	...
...	...	...
16	active, sensing, visual, intuitive	57

The learner profile for the first cluster concerning the first module, which consists of five lessons, is shown in Table 8. The provided learner ratings are presented in Table 8. The ratings were measured from a value of 1 for "marginal" to the a value of 5 for "excellent". The learner profiles were constructed for the remaining modules.

Table 8. Learner profile for Module 1 and Cluster 1.

Sequence	*Learner-Id*	Lesson 1	Lesson 2	Lesson 3	Lesson 4	Lesson 5
1	1	2	4	4	4	3
2	2	3	4	4	3	3
3	3	4	4	5	3	4
...	...	...	...	...	...	...
n	1000	5	5	3	2	5

5. Experimental Results and Analysis

In this section, we analyze a simulation of this new recommender executed on some sample datasets. The proposed method was implemented in the Java language. The proposed method was evaluated via different measurements—MAE, *Precision* (*List*, *User*), *Recall* (*List*, *User*), and *Ranking Score* (*User*).

5.1. Datasets

The proposed intelligent recommender was simulated on some educational data sets with 1000 learners. The learners were divided into two different clusters: a simulation cluster and a no-recommender cluster. The learners in the no-recommender cluster were not guided through the recommender to access the learning resources. The learners in the simulation cluster were required to go through the proposed model. The simulation cluster consisted of 900 learners and the no-recommender cluster consisted of 100 learners.

The simulation and no-recommender clusters' data accomplished the normality condition, as indicated by the statistical t-test used to check whether any additional equalization of groups was required. The experimental target was to cluster the simulation cluster learners into subclusters based on the styles of learning and characteristics of the learners. Sixteen clusters were constructed based on the different learning styles and characteristics.

The statistical t-test was conducted to test whether there were any significant differences between the means of the simulation and no-recommender clusters. The learners in these clusters completed the exercises and tests in each chapter and, hence, the intellectual skills of the learners were compared. For the n_1 = 900 learners in the simulation clusters and n_2 = 100 learners in the no-recommender clusters, the averages of the learners' intellectual skills (simulation cluster: 98.27 and no-recommender cluster: 93.73) were tested at the level of significance α = 5% with $n_1 + n_2 - 2$ degrees of freedom. The t-test resulted in a calculated value of t_{cal} = 1.57, less than t_{tab} = 1.96. This signifies that there were no significant differences between the clusters; hence, it was concluded that additional equalization of the groups was not required.

5.2. Performance Metrics

The performance of the proposed model was analyzed via different measurements. A comparison of the proposed method with the existing CF method in terms of MAE is shown in Figure 8. When the MAE became lower, the recommender predicted learner ratings more effectively. The recommended learning sequences fulfilled the accuracy requirement. The experimental results signify that the proposed method minimized the MAE metric for the different clusters considered when compared to the existing method. A significant difference was inferred in comparison with the CF method.

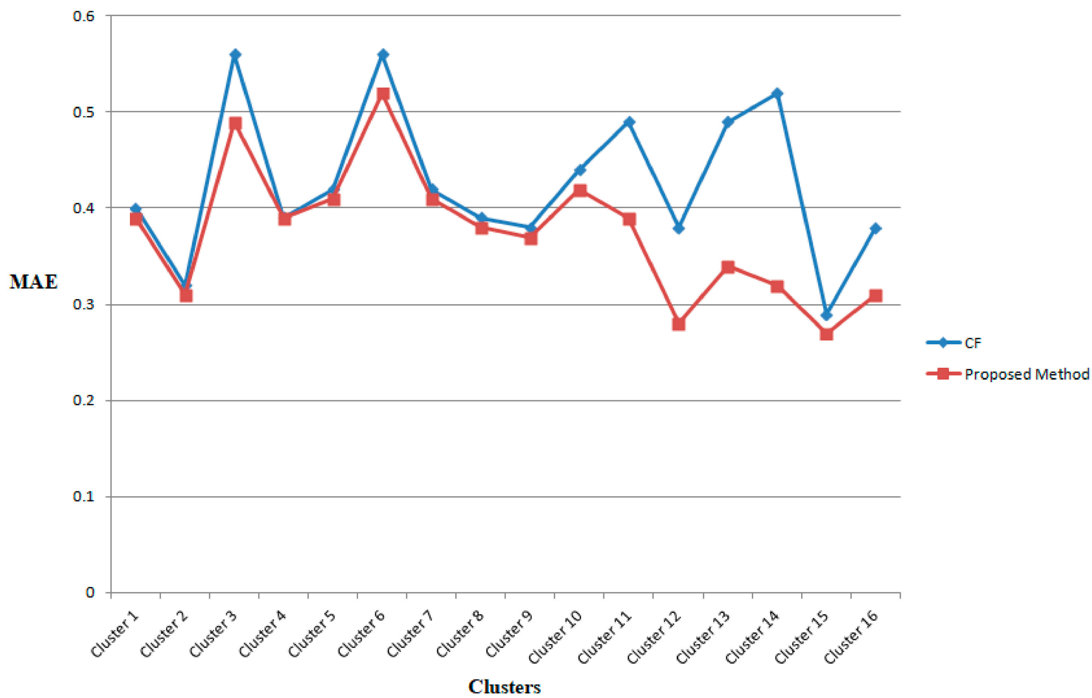

Figure 8. Performance metric comparisons—mean absolute error (MAE).

The proposed recommender's performance was compared with the existing CF using different metrics as shown in Figure 9. For higher values of *Recall* (*List*, *User*), the system gave better recommendations of the learning resources. For higher values of *Precision* (*List*, *User*), the recommender system signified that there were more items in its recommendation. For lower values of *Ranking Score* (*User*), the recommender system signified that the required item was available at the starting place.

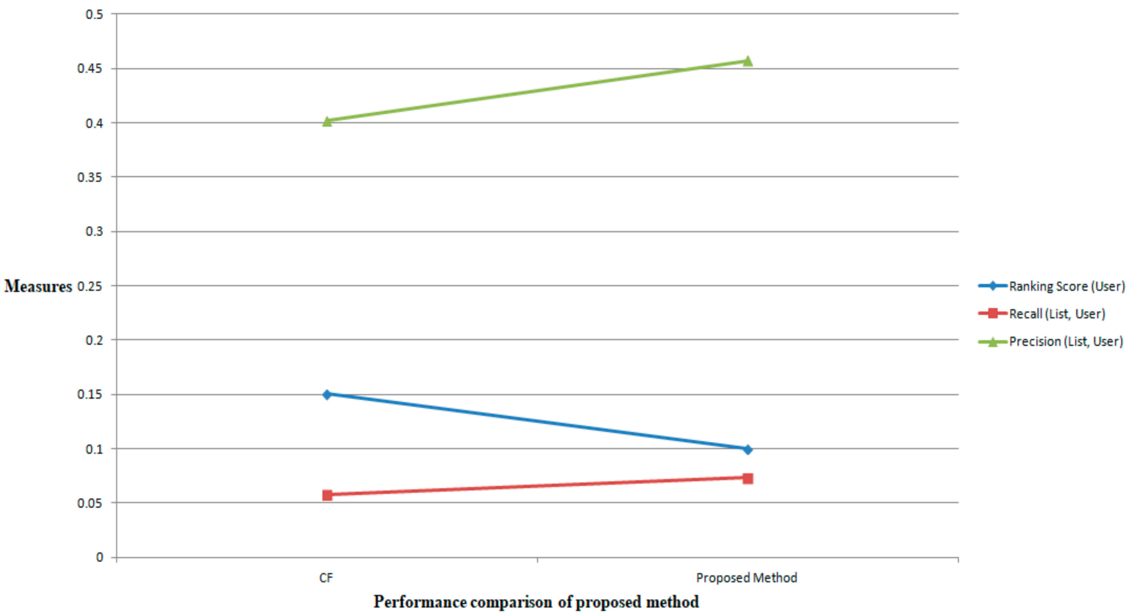

Figure 9. Performance comparisons—*Recall*, *Precision*, and *Ranking Score* metrics.

5.3. Analysis

The efficiency measure defines the time that learners need to reach the required learning goals. The devised recommender was compared with the Mass Diffusion Heat

Spreading Resource (MDHS), nearest neighborhood CF, and User Profile Oriented Diffusion (UPOD) methods. During the execution of the recommender, minimal $|L|$ was set. The experimental results showed that the learners from the simulation cluster could complete a course with reduced computational time and could complete more lessons than those from the no-recommender cluster. It was also observed that the proposed model intelligently recommended learning resources based on the characteristics and styles of learning.

The experimental results of the considered metrics were analyzed and the mean μ and standard deviation σ were calculated for the sample dataset. The simulation outcomes are tabulated in Tables 9 and 10. The results were compared using the statistical t-test to analyze the significance of the existing methods. From the data in the tables, we concluded that the proposed algorithms outperformed the existing techniques based on the metrics applied to evaluate the performance measurements. For the considered dataset, a significant difference was obtained in terms of parameters μ and σ concerning the *Ranking Score* (*User*) and *Recall* (*List*, *User*) measures in comparison with the existing methods. The proposed method also worked well in terms of all metrics in comparison to the existing methods.

Table 9. Performance comparison based on μ.

Strategy	*Ranking Score* (*User*)	*Recall* (*List*, *User*)	*Precision* (*List*, *User*)
CF	0.563	0.114	0.005
MDHS	0.282	0.292	0.182
UPOD	0.172	0.301	0.193
Proposed Method	0.070	0.326	0.216

Table 10. Performance comparison based on σ.

Strategy	*Ranking Score* (*User*)	*Recall* (*List*, *User*)	*Precision* (*List*, *User*)
CF	0.003	0.004	0.002
MDHS	0.002	0.005	0.002
UPOD	0.002	0.006	0.003
Proposed Method	0.001	0.008	0.004

The mean computational time of no-recommender versus simulation clusters is plotted in Figure 10. The lessons are plotted on the x-axis and the computational time is measured and plotted on the y-axis. The computational times were measured separately for the no-recommender and simulation clusters. It was experimentally observed that the computational time of simulation cluster learners was significantly reduced.

The expected number of completed lessons for both the no-recommender and simulation clusters is plotted in Figure 11 at different checkpoints considered in the x-axis uniformly. It was experimentally found that the proposed cluster-based recommender improved the performance, as indicated by learners in the simulation cluster completing more lessons than those in the no-recommender cluster category. The number of lessons completed on the y-axis increased over the linear scale as the checkpoint time increased on the x-axis.

The proposed hybrid intelligent recommender was evaluated based on the criteria of simplicity, speed, accuracy, and reliability of the model; this is presented in Figure 12. It was observed that more than 65% of the learners considered all criteria to evaluate the proposed recommender. The learners were also satisfied with the accuracy and speed of the recommender. The proposed strategies were thus effective in obtaining a better recommendation for learners.

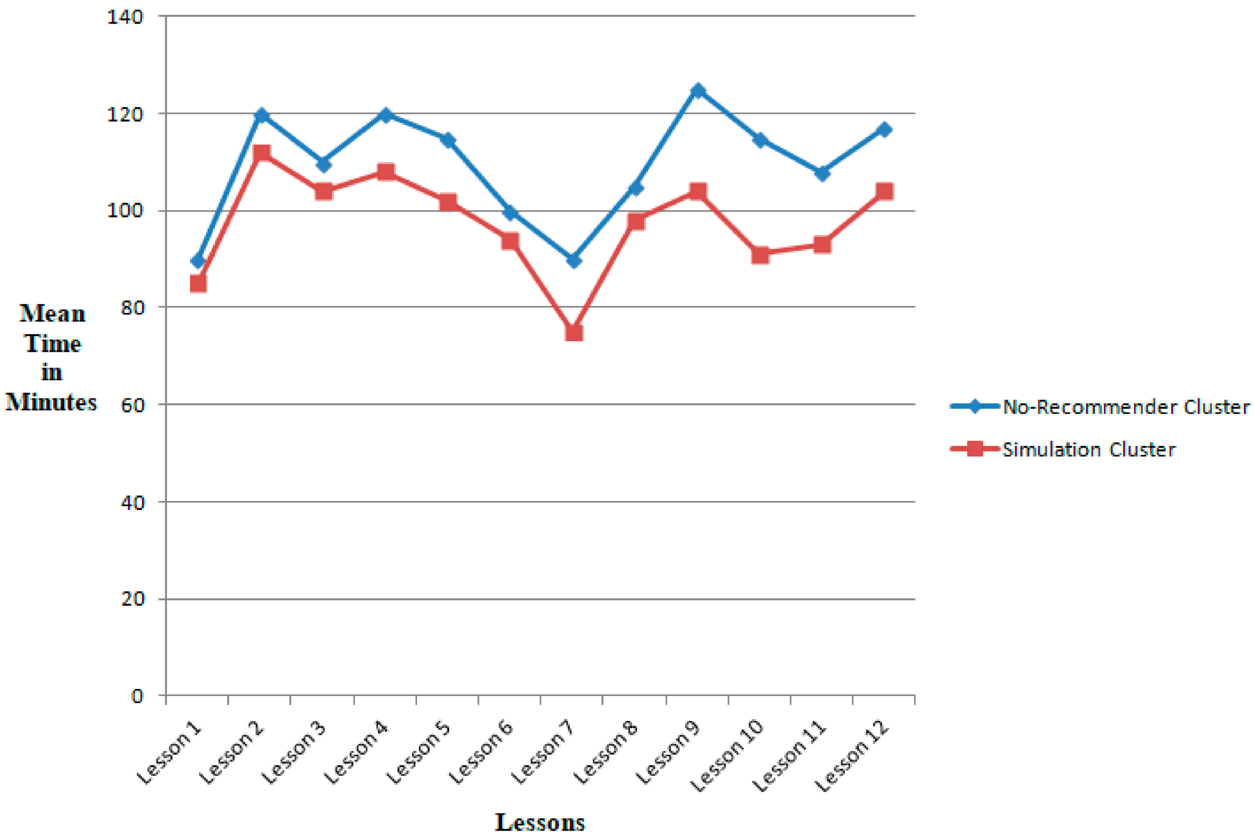

Figure 10. Mean computational time of no-recommender versus simulation clusters.

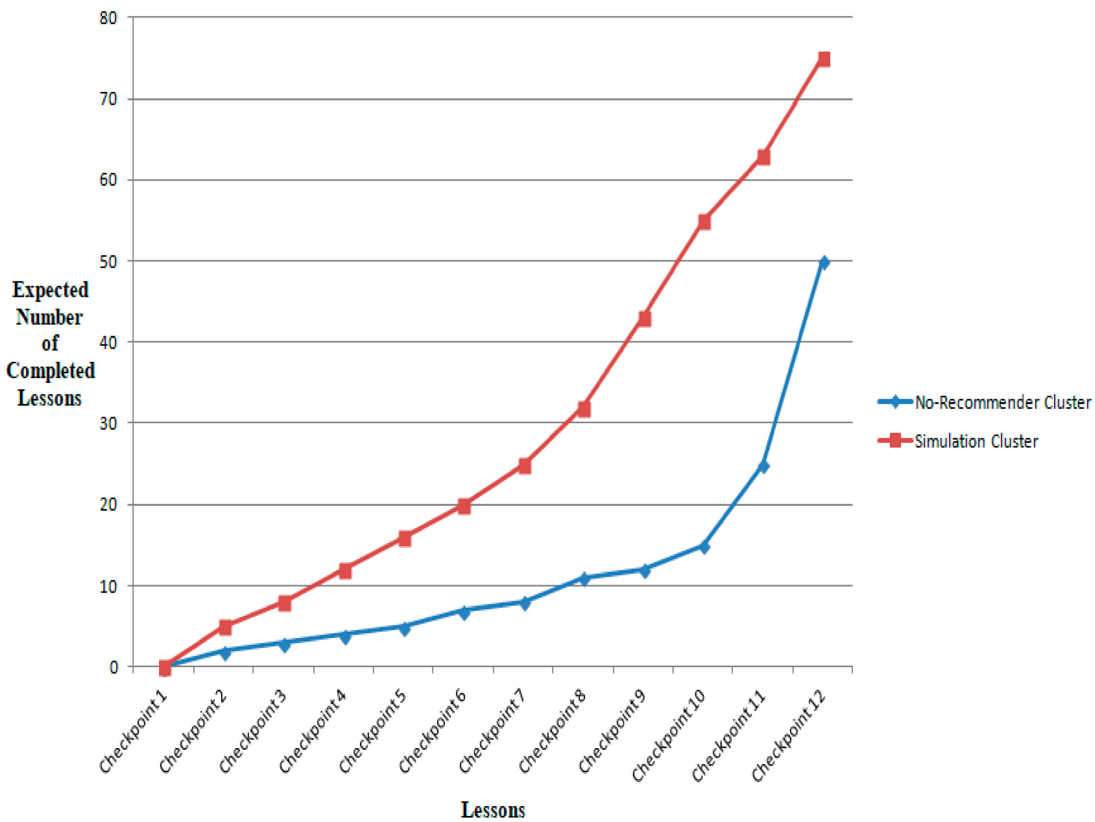

Figure 11. Expected number of completed lessons for no-recommender versus simulation clusters.

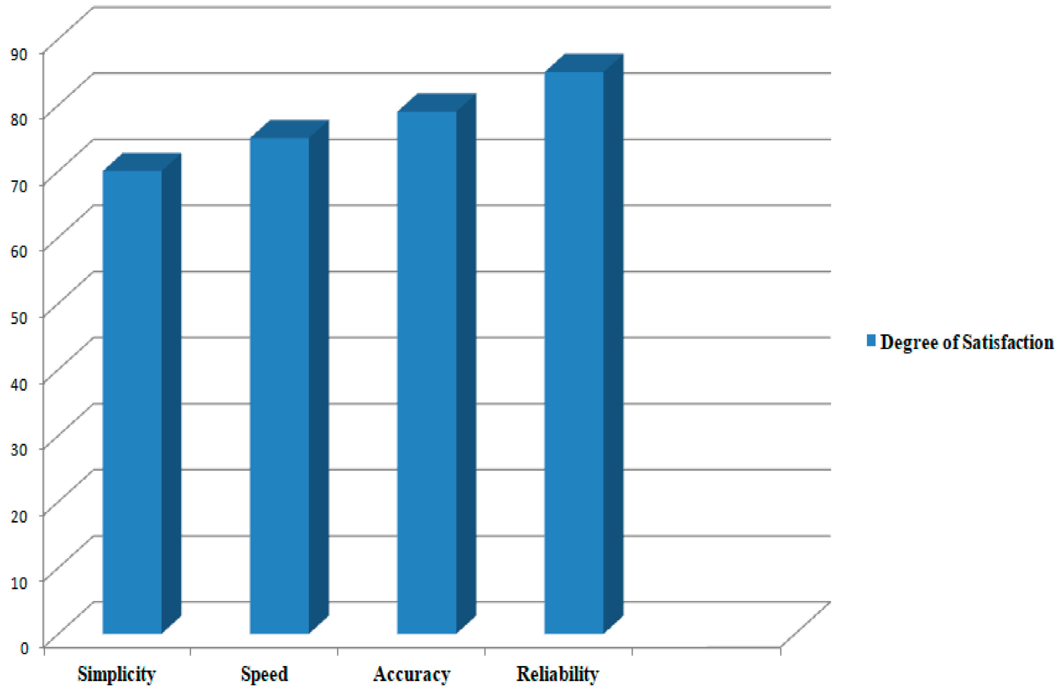

Figure 12. Evaluation of the hybrid intelligent recommender.

6. Conclusions and Future Work

Intelligent recommender systems are required for real-time e-learning applications to enhance performance. A new hybrid intelligent recommender that automatically suits the learning styles and characteristics of the learner was designed in this research. Several simulations were performed on the proposed model, and its performance measurements were compared with those of the existing CF recommender. The proposed sequential pattern clustering, pruning, and recommender algorithms produced better results compared to the existing CF. It was experimentally concluded that the proposed cluster-based recommender improved performance, as indicated by learners in the simulation cluster completing more lessons than those in the no-recommender cluster category. The simulation of the proposed recommender showed that for learner sizes of <1000, better metric values were produced. When the learner size exceeded 1000, significant differences were obtained in the evaluated metrics. The significant differences were analyzed in terms of the computational structure depending on $|L|$ and learner attributes. It was observed that more than 65% of the learners considered all criteria in evaluating the proposed recommender. The proposed method obtained upper bounds on the *Precision* and *Recall* metrics for the sample dataset of 0.326 and 0.216, respectively. The learners were also satisfied with the accuracy and speed of the recommender. The proposed strategies were thus effective in obtaining better recommendations for learners. The experimental results showed that the proposed method minimized the MAE metric for the different clusters considered when compared to the existing method. For the considered sample dataset, a significant difference was observed in the parameters standard deviation σ and mean μ concerning the *Recall* (*List*, *User*) and *Ranking Score* (*User*) measures when compared to other methods. The devised method performed well concerning all the considered metrics in comparison to the other methods. It was also found that the proposed model intelligently recommends learning resources based on the characteristics and styles of learning.

Some future research guidelines are as follows [100–105]:

- To further enhance and recommend learning resources based on the specific characteristics and learning styles of the learners;
- To apply metaheuristic strategies to further improve the performance metrics;
- To dynamically generate recommendations with minimal complexity; and

- To apply evolutionary operators and machine learning for dynamic and hybrid recommendations.

Author Contributions: Methodology, software, resources, writing—original draft and formal analysis: S.B.; conceptualization, formal analysis, validation, writing—review and editing, proofreading: R.M.; revise the paper: B.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to acknowledge the support rendered by the Management of SASTRA Deemed University in providing financial support.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Masoumi, D.; Lindström, B. Quality in e-learning: A framework for promoting and assuring quality in virtual institutions. *J. Comput. Assist. Learn.* **2012**, *28*, 27–41. [CrossRef]
2. Ossiannilsson, E.; Landgren, L. Quality in e-learning—A conceptual framework based on experiences from three international benchmarking projects. *J. Comput. Assist. Learn.* **2011**, *28*, 42–51. [CrossRef]
3. Alptekin, S.E.; Karsak, E.E. An integrated decision framework for evaluating and selecting e-learning products. *Appl. Soft Comput.* **2011**, *11*, 2990–2998. [CrossRef]
4. Sudhana, K.M.; Raj, V.C.; Zuresh, R.M. An ontology-based framework for context-aware adaptive e-learning system. In Proceedings of the International Conference on Computer Communication and Informatics, Coimbatore, India, 4–6 January 2013.
5. Kolekar, S.V.; Sanjeevi, S.G.; Bormane, D.S. Learning method recognition using artificial neural network for adaptive user interface in e-learning. In Proceedings of the IEEE International Conference on Computational Intelligence and Computing Research, Coimbatore, India, 28–29 December 2010.
6. Fernández-Gallego, B.; Lama, M.; Vidal, J.C.; Mucientes, M. Learning Analytics Framework for Educational Virtual Worlds. *Procedia Comput. Sci.* **2013**, *25*, 443–447. [CrossRef]
7. Anitha, A.; Krishnan, N. A dynamic web mining framework for E-learning recommendations using rough sets and association rule mining. *Int. J. Comput. Appl.* **2011**, *12*, 36–41. [CrossRef]
8. Keefe, J.W. *Learning Method: Theory and Practice*; National Association of Secondary School Principals: Reston, VA, USA, 1987; ISBN 0-88210-201-X.
9. Tam, V.; Lam, E.Y.; Fung, S.T. Toward a complete e-learning system framework for semantic analysis, concept clustering and learning path optimization. In Proceedings of the IEEE 12th International Conference on Advanced Learning Technologies, Rome, Italy, 4–6 July 2012.
10. Ghauth, K.I.; Abdullah, N.A. Learning materials recommendation using good learners' ratings and content-based filtering. *Educ. Technol. Res. Dev.* **2010**, *58*, 711–727. [CrossRef]
11. Verbert, K.; Manouselis, N.; Ochoa, X.; Wolpers, M.; Drachsler, H.; Bosnic, I.; Duval, E. Context-aware recommender systems for learning: A survey and future challenges. *IEEE Trans. Learn. Technol.* **2012**, *5*, 318–335. [CrossRef]
12. Hsu, M.H. A personalized English learning recommender system for ESL students. *Expert Syst. Appl.* **2008**, *34*, 683–688. [CrossRef]
13. Lu, J. *A Personalized E-Learning Material Recommender System. International Conference on Information Technology and Applications*; Macquarie Scientific Publishing: Sydney, Australia, 2004.
14. Chen, C.-M.; Lee, H.-M.; Chen, Y.-H. Personalized e-learning system using Item Response Theory. *Comput. Educ.* **2005**, *44*, 237–255. [CrossRef]
15. Abel, F.; Gao, Q.; Houben, G.J.; Tao, K. Analyzing User Modeling on Twitter for Personalized News Recommendations. In *Proceedings of the International Conference on User Modeling Adaptation, and Personalization*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 1–12.
16. Bachari, E.E.; Abelwahed, E.H.; Adnani, M. E-learning personalization based on dynamic learners'preference. *Int. J. Comput. Sci. Inf. Technol. (IJCSIT)* **2011**, *3*, 200–216.
17. Gallego, D.; Barra, E.; Aguirre, S.; Huecas, G. A model for generating proactive context-aware recommendations in e-learning systems. In Proceedings of the 2012 Frontiers in Education Conference Proceedings, Seattle, WA, USA, 3–6 October 2012.
18. Yu, Z.; Nakamura, Y.; Jang, S.; Kajita, S.; Mase, K. Ontology-based semantic recommendation for context-aware e-learning. In *Proceedings of the International Conference on Ubiquitous Intelligence and Computing*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 898–907.

19. Tai, D.W.S.; Wu, H.J.; Li, P.H. Effective e-learning recommendation system based on self-organizing maps and association mining. *Electron. Libr.* **2008**, *26*, 329–344.
20. Zhou, Y.; Huang, C.; Hu, Q.; Zhu, J.; Tang, Y. Personalized learning full-path recommendation model based on LSTM neural networks. *Inf. Sci.* **2018**, *444*, 135–152. [CrossRef]
21. Nafea, S.M.; Siewe, F.; He, Y. A novel algorithm for course learning object recommendation based on student learning styles. In Proceedings of the IEEE International Conference on Innovative Trends in Computer Engineering (ITCE), Aswan, Egypt, 2–4 February 2019; pp. 192–201.
22. Doja, M.N. An Improved Recommender System for E-Learning Environments to Enhance Learning Capabilities of Learners. In *Proceedings of the ICETIT 2019*; Springer: Cham, Switzerland, 2020; pp. 604–612.
23. Adomavicius, G.; Tuzhilin, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 734–749. [CrossRef]
24. Agrawal, R.; Srikant, R. Mining sequential patterns. In Proceedings of the 11th International Conference on Data Engineering, Taipei, Taiwan, 6–10 March 1995; pp. 3–14.
25. Cha, H.J.; Kim, Y.S.; Park, S.H.; Yoon, T.B.; Jung, Y.M.; Lee, J.H. Learning style diagnosis based on user interface behavior for the customization of learning interfaces in an intelligent tutoring system. Lecture notes in computer science. In *Proceedings of the International Conference on Intelligent Tutoring Systems*; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4053, pp. 513–524.
26. Coffield, F.; Moseley, D.; Hall, E.; Ecclestone, K. *Should We Be Using Learning Styles? What Research has to Say to Practice. Report of the Learning and Skills*; Development Agency, Regent Arcade House: London, UK, 2004. Available online: http://www.lsneducation.org.uk/research/reports/ (accessed on 14 June 2010).
27. Dikovi, L. Applications GeoGebra into teaching some topics of mathematics at the college level. *Comput. Sci. Inf. Syst.* **2009**, *6*, 191–203. [CrossRef]
28. Drachsler, H.; Hummel, H.G.K.; Koper, R. Identifying the goal, user model and conditions of recommender systems for formal and informal learning. *J. Digit. Inf.* **2009**, *10*, 4–24.
29. Dunn, R.; Dunn, K.; Freeley, M.E. Practical applications of the research: Responding to students' learning styles-step one. *Ill. State Res. Dev. J.* **1984**, *21*, 1–21.
30. Emurian, H. A web-based tutor for Java: Evidence of meaningful learning. *Int. J. Distance Educ. Technol.* **2006**, *4*, 10–30. [CrossRef]
31. Farzan, R.; Brusilovsky, P. Social navigation support in a course recommendation system. In Proceedings of the 4th International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, Dublin, Ireland, 21–23 June 2006; pp. 91–100.
32. Felder, R.M.; Silverman, L.K. Learning and teaching styles in engineering education. *Eng. Educ.* **1988**, *78*, 674–681.
33. Felder, R.M.; Soloman, B.A. Index of Learning Styles Questionnaire. 1996. Available online: http://www.engr.ncsu.edu/learningstyles/ilsweb.html (accessed on 17 December 2009).
34. Fertalj, K.; Bozic-Hoic, N.; Jerkovic, H. The integration of learning object repositories and learning management systems. *Comput. Sci. Inf. Syst.* **2010**, *7*, 387–407. [CrossRef]
35. García, P.; Amandi, A.; Schiaffino, S.; Campo, M. Evaluating Bayesian networks' precision for detecting students' learning styles. *Comput. Educ.* **2007**, *49*, 794–808. [CrossRef]
36. Glaser, R. Instructional technology and the measurement of learning outcomes. *Am. Psychol.* **1963**, *18*, 510–522. [CrossRef]
37. Graf, S.; Kinshuk, K. Providing adaptive courses in learning management systems with respect to learning styles. In *Proceedings of the World Conference on E-Learning in Corporate Government, Healthcare, and Higher Education*; Bastiaens, T., Carliner, S., Eds.; AACE: Chesapeake, VA, USA, 2007; pp. 2576–2583.
38. Graf, S.; Liu, T.C. Analysis of learners' navigational behavior and their learning styles in an online course. *J. Comput. Assist. Learn.* **2010**, *26*, 116–131. [CrossRef]
39. Hammouda, K.; Kamel, M. Data mining in e-learning. In *E-Learning Networked Environments and Architectures: A Knowledge Processing Perspective*; Pierre, S., Ed.; Springer Book Series: Advanced Information and Knowledge Processing; Springer: London, UK, 2006; pp. 374–402.
40. Herlocker, J.; Konstan, J.; Borchers, A.; Riedl, J. An algorithmic framework for performing collaborative filtering. *Proc. ACM SIGIR Conf.* **1999**, *99*, 230–237.
41. Herlocker, J.L.; Konstan, J.A.; Terveen, L.G.; Riedl, J.T. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* **2004**, *22*, 5–53. [CrossRef]
42. Holland, J.; Mitrovic, A.; Martin, B. J-Latte: A constraint-based tutor for java. In Proceedings of the 17th International Conference on Computers in Education, Hong Kong, China, 30 November–4 December 2009; pp. 142–146.
43. Honey, P.; Mumford, A. *The Manual of Learning Styles*; Peter Honey: Maidenhead, UK, 1982.
44. Ivanovi, M.; Pribela, I.; Vesin, B.; Budimac, Z. Multifunctional environment for E-learning purposes. *Novi Sad J. Math. NSJOM* **2008**, *38*, 153–170.
45. Janssen, J.; Van den Berg, B.; Tattersall, C.; Hummel, H.; Koper, R. Navigational support in lifelong learning: Enhancing effectiveness through indirect social navigation. *Interact. Learn. Environ.* **2007**, *15*, 127–136. [CrossRef]
46. Kelly, D.; Tangney, B. *Prediciting Learning Characteristics in a Multiple Intelligence Based Tutoring System*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 9–30.
47. Klašnja-Milićević, A.; Vesin, B.; Ivanovic, M.; Budimac, Z. Integration of recommendations and adaptive hypermedia into java tutoring system. *Comput. Sci. Inf. Syst.* **2011**, *8*, 211–224. [CrossRef]

48. Kolb, D. Individuality in learning and the concept of learning styles. In *Experiential Learning*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1984; pp. 61–98.
49. Koper, E.J.R. Increasing learner retention in a simulated learning network using indirect social interaction. *J. Artif. Soc. Soc. Simul.* **2005**, *8*, 18–27.
50. Kristofic, A. Recommender system for adaptive hypermedia Applications. In Proceedings of the Informatics and Information Technology Student Research Conference, Bratislava, Slovakia, 27 April 2005; pp. 229–234.
51. Linden, G.; Smith, B.; York, J. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Comput.* **2003**, *7*, 76–80. [CrossRef]
52. Myers, I.B. *Manual: The Myers-Briggs Type Indicator*; Consulting Psychologists Press: Palo Alto, CA, USA, 1962.
53. Northrup, P.A. Framework for designing interactivity into web-based instruction. *Educ. Technol.* **2001**, *41*, 31–39.
54. Parvez, S.M.; Blank, G.D. Individualizing tutoring with learning style based feedback. In Proceedings of the Intelligent Tutoring Systems 9th International Conference, Montreal, QC, Canada, 23–27 June 2008; Springer: Berlin/Heidelberg, Germany, 2008; pp. 291–301.
55. Pashler, H.; McDaniel, M.; Rohrer, D.; Bjork, R. Learning styles: Concepts and evidence. *Psychol. Sci. Public Interest* **2009**, *9*, 105–119. [CrossRef] [PubMed]
56. Pask, G. Styles and strategies of learning. *Br. J. Educ. Psychol.* **1976**, *46*, 128–148. [CrossRef]
57. Peña, C.I.; Marzo, J.L.; de la Rosa, J.L. Intelligent agents in a teaching and learning environment on the web. In Proceedings of the IEEE International Conference on Advanced Learning Technologies, Kazan, Russia, 9–12 August 2002; pp. 21–27.
58. Popescu, E. Dynamic Adaptive Hypermedia Systems for E-Learning. Ph.D. Thesis, Université de Craiova, Craiova, Romania, 2008.
59. Resnick, P.; Iacovou, N.; Suchak, M.; Bergstrom, P.; Riedl, J. GroupLens: An open architecture for collaborative filtering of netnews. In Proceedings of the ACM Conference on Computer Supported Cooperative Work, Chapel Hill, NC, USA, 22–26 October 1994; pp. 175–186.
60. Romero, C.; Ventura, S.; De Bra, P. Knowledge discovery with genetic programming for providing feedback to courseware. *User Modeling User Adapt. Interact.* **2004**, *14*, 425–464. [CrossRef]
61. Romero, C.; Ventura, S.; De Bra, P.; De Castro, C. Discovering Prediction Rules in AHA! Courses. In Proceedings of the User Modeling Conference, Johnstown, PA, USA, 22–26 June 2003; pp. 35–44.
62. Schafer, J.B.; Konstan, J.; Riedl, J. Recommender Systems in E-commerce. In Proceedings of the 1st ACM Conference on Electronic Commerce, Denver, CO, USA, 3–5 November 1999; pp. 158–166.
63. Shardanand, U.; Maes, P. Social information filtering: Algorithms for automating 'Word of Mouth'. In Proceedings of the SIGCHI Conference '95, Denver, CO, USA, 7–11 May 1995; pp. 210–217.
64. Sykes, E.R.; Franek, F. An intelligent tutoring system prototype for learning to program java. In Proceedings of the third IEEE International Conference on Advanced Learning Technologies (ICALT'03), Athens, Greece, 9–11 July 2003; pp. 485–492.
65. Tang, T.Y.; McCalla, G. Smart recommendation for an evolving e-learning system: Architecture and experiment. *Int. J. e-Learn.* **2005**, *4*, 105–129.
66. Tong, W.; Pi-Lian, H. Web log mining by an improved Aprioriall algorithm. *Eng. Technol.* **2005**, *4*, 97–100.
67. Tseng, J.C.R.; Chu, H.C.; Hwang, G.J.; Tsai, C.C. Development of an adaptive learning system with two sources of personalization information. *Comput. Educ.* **2008**, *51*, 776–786. [CrossRef]
68. Vesin, B.; Ivanovi, M.; Budimac, Z. Learning management system for programming in java. *Ann. Univ. Sci. Rolando Eötvös Nomin. Sect. Comput.* **2009**, *31*, 75–92.
69. Vesin, B.; Ivanovi, M.; Budimac, Z.; Pribela, I. MILE—Multifunctional integrated learning environment. In Proceedings of the IADIS Multi Conference on Computer Science and Information Systems MCCSIS'2008, Amsterdam, The Netherlands, 22–27 July 2008; pp. 104–108.
70. Wang, H.C.; Li, T.Y.; Chang, C.Y. A web based tutoring system with styles matching strategy for learning spatial geometry. In Proceedings of the International Computer Symposium, Taipei, Taiwan, 15–17 December 2004; pp. 226–233.
71. Wolf, C. iWeaver: Towards 'learning style'—Based e-learning in computer science education. In Proceedings of the Australasian Computing Education Conference, Adelaide, Australia, 29 January–1 February 2013; Volume 20.
72. Zaïane, O.R. Building a recommender agent for e-learning systems. In Proceedings of the International Conference on Computers in Education, ICCE'02, Auckland, New Zealand, 3–6 December 2002; pp. 55–59.
73. Zhang, J.; Wang, Y.; Yuan, Z.; Jin, Q. Personalized real-time movie recommendation system: Practical prototype and evaluation. *Tsinghua Sci. Technol.* **2020**, *25*, 180–191. [CrossRef]
74. Bertani, R.M.; Bianchi, R.A.; Costa, A.H.R. Combining novelty and popularity on personalised recommendations via user profile learning. *Expert Syst. Appl.* **2020**, *146*, 113–149. [CrossRef]
75. Shu, J.; Shen, X.; Liu, H.; Yi, B.; Zhang, Z. A content-based recommendation algorithm for learning resources. *Multimedia Syst.* **2018**, *24*, 163–173. [CrossRef]
76. Syed, M.A.; Rakesh, K.L.; Gopal, K.N.; Rabindra, K.B. Movie recommendation system using genome tags and content-based filtering. In *Advances in Data and Information Sciences*; Springer: Singapore, 2018; pp. 85–94. [CrossRef]
77. Klašnja-Milićević, A.; Vesin, B.; Ivanović, M.; Budimac, Z. E-Learning personalization based on hybrid recommendation strategy and learning style identification. *Comput. Educ.* **2011**, *56*, 885–899. [CrossRef]

78. Cui, G.; Luo, J.; Wang, X. Personalized travel route recommendation using collaborative filtering based on GPS trajectories. *Int. J. Digit. Earth* **2018**, *11*, 284–307. [CrossRef]
79. Juliana, A.P.; Pawel, M.; Sebastian, K.; Myra, S.; Gunter, S. A Feature-based personalized recommender system for product-line configuration. In Proceedings of the 2016 ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences, Amsterdam, The Netherlands, 20 October 2016; pp. 120–131. [CrossRef]
80. Liu, R.R.; Jia, C.X.; Zhou, T.; Sun, D.; Wang, B.H. Personal recommendation via modified collaborative filtering. *Physica A* **2009**, *388*, 462–468. [CrossRef]
81. Jinpeng, C.; Yu, L.; Deyi, L. Dynamic group recommendation with modified collaborative filtering and temporal factor. *Int. Arab J. Inf. Technol.* **2016**, *13*, 294–301.
82. Chaturvedi, A.; Green, P.; Caroll, J. K-modes clustering. *J. Classif.* **2001**, *18*, 35–55. [CrossRef]
83. Kourosh, M. Recommendation system based on complete personalization. *Procedia Comput. Sci.* **2016**, *80*, 2190–2204. [CrossRef]
84. Tian, Y.; Zheng, B.; Wang, Y.; Zhang, Y.; Wu, Q. College library personalized recommendation system based on hybrid recommendation algorithm. *Procedia CIRP* **2017**, *83*, 490–494. [CrossRef]
85. Marchela, A.; Christos, C. Personalized micro-service recommendation system for online news. *Procedia Comput. Sci.* **2019**, *160*, 610–615. [CrossRef]
86. Julián, M.-P.; Jose, A.; Edwin, M.; Camilo, S. Autonomous recommender system architecture for virtual learning environments. *Appl. Comput. Inf.* **2020**. [CrossRef]
87. Anand, S.T. Generating items recommendations by fusing content and user-item based collaborative filtering. *Procedia Comput. Sci.* **2020**, *167*, 1934–1940. [CrossRef]
88. Pradeep, K.R.; Sarabjeet, S.C.; Rocky, B. A machine learning approach for automation of resume recommendation system. *Procedia Comput. Sci.* **2020**, *167*, 2318–2327. [CrossRef]
89. Hanane, Z.; Souham, M.; Chaker, M. New contextual collaborative filtering system with application to personalized healthy nutrition education. *J. King Saud. Univ. Comput. Inf. Sci.* **2020**. [CrossRef]
90. Christian, R.; Michael, W.; Günther, P. State of the art of reputation-enhanced recommender systems. *Web Intell.* **2018**, *16*, 273–286.
91. Pasquale, D.M.; Lidia, F.; Fabrizio, M.; Domenico, R.; Giuseppe, M.L.S. Providing recommendations in social networks by integrating local and global reputation. *Inform. Syst.* **2018**, *78*, 58–67. [CrossRef]
92. Barry, S.; Maurice, C.; Peter, B.; Kevin, M.; Michael, P.O. Collaboration, Reputation and Recommender Systems in Social Web Search. In *Recommender Systems Handbook*; Springer: Boston, MA, USA, 2015; pp. 569–608. [CrossRef]
93. Stai, E.; Kafetzoglou, S.; Tsiropoulou, E.E.; Papavassiliou, S. A holistic approach for personalization, relevance feedback & recommendation in enriched multimedia content. *Multimed. Tools Appl.* **2018**, *77*, 283–326. [CrossRef]
94. Vasiliki, P.; Stella, K.; Eirini, E.T.; Aggeliki, D.; Symeon, P. Personalized multimedia content retrieval through relevance feedback techniques for enhanced user experience. In Proceedings of the 13th International Conference on Telecommunications (ConTEL), Graz, Austria, 13–15 July 2015. [CrossRef]
95. Simon, D. Dynamic Generation of Personalized Hybrid Recommender Systems. In Proceedings of the 7th ACM Conference on Recommender Systems, RecSys'13, Hong Kong, China, 12–16 October 2013; pp. 443–446. [CrossRef]
96. Thai, M.T.; Wu, W.; Xiong, H. *Big Data in Complex and Social Networks*, 1st ed.; Chapman & Hall/CRC Big Data Series; Chapman & Hall/CRC Press: Boca Raton, FL, USA, 2016; ISBN 978-1498726849.
97. Abbas, S.M.; Alam, K.A.; Shamshirband, S. A soft-rough set based approach for handling contextual sparsity in context-aware video recommender systems. *Mathematics* **2019**, *7*, 740. [CrossRef]
98. Sardianos, C.; Ballas Papadatos, G.; Varlamis, I. Optimizing parallel collaborative filtering approaches for improving recommendation systems performance. *Information* **2019**, *10*, 155. [CrossRef]
99. Pajuelo-Holguera, F.; Gómez-Pulido, J.A.; Ortega, F. Performance of two approaches of embedded recommender systems. *Electronics* **2020**, *9*, 546. [CrossRef]
100. Bai, L.; Hu, M.; Ma, Y.; Liu, M. A hybrid two-phase recommendation for group-buying e-commerce applications. *Appl. Sci.* **2019**, *9*, 3141. [CrossRef]
101. Cintia Ganesha Putri, D.; Leu, J.-S.; Seda, P. Design of an unsupervised machine learning-based movie recommender system. *Symmetry* **2020**, *12*, 185. [CrossRef]
102. Bhaskaran, S.; Santhi, B. An efficient personalized trust based hybrid recommendation (TBHR) strategy for e-learning system in cloud computing. *Cluster. Comput.* **2019**, *22*, 1137–1149. [CrossRef]
103. Marappan, R.; Sethumadhavan, G. Solution to graph coloring using genetic and tabu search procedures. *Arab. J. Sci. Eng.* **2018**, *43*, 525–542. [CrossRef]
104. Marappan, R.; Sethumadhavan, G. Complexity analysis and stochastic convergence of some well-known evolutionary operators for solving graph coloring problem. *Mathematics* **2020**, *8*, 303. [CrossRef]
105. Bhaskaran, S.; Marappan, R.; Santhi, B. Design and Comparative Analysis of New Personalized Recommender Algorithms with Specific Features for Large Scale Datasets. *Mathematics* **2020**, *8*, 1106. [CrossRef]