

Interactive Hybrid Recommendation of Pedagogical Resources

Chahrazed Mediani

Computer Science Department and LRSD Laboratory, University Ferhat Abbes of Setif, Setif 19000, Algeria

Corresponding Author Email: chahrazed.mediani@univ-setif.dz



<https://doi.org/10.18280/isi.270502>

ABSTRACT

Received: 17 April 2022

Accepted: 30 August 2022

Keywords:

collaborative filtering, content-based, recommender system, pedagogical resource, SVM, TF-IDF

Recommender systems provide a solution to the information overload problem and can estimate user preferences in a given resource from certain information about other similar users and the resource's features. This research presented a recommender system for pedagogical resources. The built system adapts a hybrid method of two principal approaches: content-based and collaborative filtering. We have used the term frequency-inverse document frequency (TF-IDF) technique and singular value decomposition (SVD) to let the system better understand the semantics of the pedagogical resources in a content-based approach and prevent some shortcomings of the existing collaborative filtering approach. The hybridization of those techniques and methods helps the system improve the accuracy of recommendations and better respond to user requirements. The first tests of our system give encouraging results.

1. INTRODUCTION

Nowadays, a large mass of information is available on the Internet. It has become essential to design mechanisms that allow users to access as quickly as possible what they want. Recommendation systems (RS) provide a solution to the problem of information overload, considering the heart of modern information systems that we use daily [1]. These systems have played a vital and indispensable role in various information access systems to drive business and facilitate decision-making. A recommender system is a specific form of information filtering that helps users find items from massive catalogues to present them based on their behavior and preferences [2, 3]. Items can be books, movies, news, music, and others. It also helps users make choices in an area where they have little information to sort and evaluate possible alternatives.

Recommender systems are applications that can filter information in a personalized manner [4]; the filtering algorithm characterizes the internal functions of recommendation systems. Several classifications of RS have emerged [4]. These are based on different factors to categorize a recommender system: knowing the user (i.e., his profile), knowing the user position between the others, knowing the items that we will recommend, and knowing in which class is the item we want to recommend. The most widely used classification divides filtering algorithms into popularity-based (PB), content-based (CB), collaborative filtering (CF), and hybrid-based recommender system (HB), which is the hybridization of them.

Popularity-based recommender system is the simplest system compared to other more advanced approaches. Popularity-based stands for the principle of the public endorsement considerably affects others' choices or decision-making. There is a big chance for users to change their pre-made choices once they deal with popular items [5]. The popularity-based recommender system recommends popular

items, among others, due to the capacity of the interactions many users provide regarding those items [6]. However, those systems give non-personal recommendations because they suggest most users' preferences [7].

Content-based recommender systems are based on user ratings of a set of documents or items [8]. The purpose is to understand the motivations that led him to judge a given item as relevant or not, then offer a choice among new items to be close to the items he has previously enjoyed [9]. A content-based approach analyzes a set of items that have been rated or viewed previously [10]. It builds a profile or a model of the user's interests based on the characteristics of items liked or disliked by the user. Based on feedback, the user's profile is constructed. The recommendation process compares the candidate's attributes with the user's attributes profile. And as a final result, the system will recommend items similar to the user profile.

Collaborative filtering is the basis of the recommendation. Systems based on collaborative filtering consider user ratings of items to calculate the similarity of their preferences and recommend these items without analyzing their content [11]. It is no longer based on the notion of proximity of a "new item - user profile" pair but seeks to bring the current user closer to a set of existing users. The idea here is no longer to be specifically interested in the new item that would be likely to please the user but to look at which items liked the users close to the current user. All users of the collaborative filtering system can take advantage of the reviews of others by receiving recommendations for which the closest users have made a favorable value judgment without the system having content extraction process documents. Thanks to its independence, this technique can be used for any data: text, image, audio, and video. There are two main classes of collaborative filtering: memory-based and model-based [12].

The memory-based (MB) approach predicts ratings based on users' previously rated items [12]. MB methods generally use similarity metrics user-wise or item-wise to predict

recommendations. For example, suppose the system adapts the user-wise memory-based approach. In that case, the system will determine the relationship between a particular user u and the rest of the users $u_i, i \in U$, by calculating the distance based on their interacted items: $sim(u, u_i)$. Thus, a user will get recommendations from other similar users. Similarity measures estimate ratings by grouping users into groups of closest peers. The most common similarity measure is cosine-similarity, where users are represented as vectors in an n -dimensional space. Thus, the similarity is calculated between users as the cosine of the angle between them. Memory-based algorithms differ from the context they are used in; for example, in the content-based approach, similarity measure is between vectors of TF-IDF weights, whereas in the collaborative approach, it measures the similarity between vectors of the actual user-specified ratings [12].

Unlike memory-based, model-based methods include machine learning approaches to utilize the data and generate a learned model to explain the user-item interactions and discover it to make new predictions [12]. The developed model is trained to recreate the user-item interaction values. The recommendations are based on this trained model by extracting latent representations of users and items and learning their mathematical meaning. MF is the most know model-based approach [6]. MF algorithms break the sparse user-item interaction matrix into a product of two smaller and dense matrices: users representations in a user-factor matrix and items representations in a factor-item matrix [12].

The above recommender framework approaches have shown to be exceptionally solid and dependable. Nonetheless, they face a few weaknesses and shortcomings, where hybrid frameworks are acquainted with further development suggestion frameworks. A hybrid framework is a blend of at least two principle approaches [2]. The primary idea behind a hybrid framework is to fill the shortcoming of one methodology with the qualities of the others [2]. A Hybrid recommender system implementation requires specifying a hybridization strategy. A taxonomy for classifying hybrid recommender systems is proposed by Burke [13]. They are divided into seven classes: weighted, switching, cascade, feature augmentation, feature combination, meta-level, and mixed hybrids. Therefore, based on the workflow, hybrid systems are organized into two main categories: Palatalized hybrid design (Mixed, Weighted, Switched), where the selected approaches work parallelly at the same time, and Pipelined hybrid design (Cascade, Meta-level), where the main recommender approaches use other approaches as a baseline [2, 14].

Recommender frameworks are fundamental in the industry, mainly trading and marketing, to support benefits [11]. For instance, a web-based e-commerce platform tries to satisfy its clients and outflank contenders by giving the main recommendations and services [11]. Nonetheless, the effect of recommender frameworks likewise arrived at different fields, like social media applications, to ensure the best user navigation experience, scientific fields, and educational purposes. For instance, students get suggestions of educational resources according to their needs and preferences in e-learning platforms [15] and different fields. Regardless of what the industries advantage from recommender frameworks, users altogether benefit from recommender frameworks regarding users not dealing with the static type of information. Instead, they utilize an interactive approach to introduce their preferences, implying that users have unlimited authority over

their data [1]. RS faces a few challenges in addition to the effect and improvement of recommendation systems regarding user navigation experience and information overpowering. Existing recommender frameworks come up short on expanded volume, heterogeneity, and the reality that they are not adequately adjusted to the user's requirements and needs.

As of late, recommender frameworks have begun including machine learning productivity to get users' way of behaving and increment recommendations accuracy. Among online platforms, such as e-commerce and social media platforms, online educational platforms (E-learning) also attempt to profit from recommender systems to work with and further develop learning. Those educational platforms give users (teachers and learners) the proper pedagogical devices to assist them with building a superior educational environment, including collaboration and assets sharing.

The research found that combining content-based and collaborative filtering creates a separate system [16]. Such a system is known to be a hybrid recommendation system with both the features of content-based filtering and a collaborative filtering system [17]. It considers the users' likes to adjust the system to the level of preferences held by the user.

Recommender systems play an essential role in the industry, especially trading and marketing, to boost profits. For example, a web-based e-commerce platform seeks to please its customers and outperform competitors by providing the most significant recommendations and services. However, the impact of recommender systems also reached other fields, such as social media applications, to guarantee the best user navigation experience, scientific fields, and educational purposes. For example, students get recommendations of educational resources according to their preferences in e-learning platforms and other areas [15]. Regardless of the benefits of industries from recommender systems, learners significantly benefit from recommender systems in terms of learners not dealing with the static form of data. Instead, they use an interactive approach to define their preferences, which means the learners have complete control of their data [1]. So, given a large number of pedagogical resources, learners can lose their points of reference: what resources to consult first? Therefore, learners can feel the need to be supported by services enabling them to access relevant resources to reach their goals. In the framework of our work, we are interested in using recommender systems within learning platforms to help learners access pedagogical resources serving their goals. A pedagogical resource can have different types: a website, a book, course support, a serious game, a person, a response to a question, etc.

Our research proposed a hybrid recommender system for pedagogical resources. This system combined a content-based approach and a collaborative filtering approach. In the content-based approach, we decided to use term frequency-inverse document frequency (TF-IDF) [18] for those three reasons: (1) a content-based recommender system does not require domain knowledge. Only the user knowledge, (2) Immediate consideration of a new item: content-based filtering can recommend newly introduced items in the database even before they receive an evaluation from a user, and (3) content-based recommendation techniques treat each user independently.

For the collaborative filtering approach, we used the Singular Value Decomposition (SVD) latent factor models [19]. One advantage of this approach is that instead of a high dimension matrix with a huge number of missing values, we

will deal with a much smaller matrix in a smaller space. A reduced presentation can be used for neighboring algorithms based on users or items. This paradigm's advantages are that it handles the utility matrix's sparsity better than memory-based. In addition, the comparison of similarities with the resulting matrix is much more ambitious, especially when dealing with large sparse data sets.

We summarized our contributions as follows:

- We developed a hybrid recommender system for pedagogical resources (HrecSys). The HrecSys framework used a hybridization of two recommender system approaches: content-based and collaborative filtering. HrecSys framework is constructed based on mapping the original data to higher-order features interactions. This architecture aims to reduce the user-item interactions bias and improve the effectiveness of the content-based and collaborative systems. Furthermore, the HrecSys framework falls in both mixing classes of Burke's taxonomy, as it can use a combination of its underlying sub-recommenders, all according to their use context.

- The proposed HrecSys used TF-IDF technology to let the system better understand the mathematical significance of words of pedagogical resources in the content-based approach. In contrast, we used a SVD matrix factorization in collaborative filtering to predict users' ratings better. The system also applies a mixed strategy using a weighted score of content-based and collaborative filtering. It provides a flexible architecture allowing the creation of various instances.

- To evaluate HrecSys, a hybrid instance is developed. It is composed of two innovative sub-recommenders: CB-TFIDF and CF-SVD.

- The CB-TFIDF exploits the strengths of embedding representations for modelling the content. This recommender aims to extract effective features from the item-based side information. Which would ameliorate the analysis of user-item interactions and thus result in more accurate predictions.

- The CF-SVD recommender attempts to model a high level of non-linearities and displays interactions between users and items in latent embeddings. It can also reduce users' biases towards items rated by users.

- The HrecSys framework can be used as a model for combining different sub-recommenders. According to our assessments, the HrecSys instance outperforms state-of-the-art recommendation algorithms on various real-world datasets.

In the following, we state our problem and present existing work related to hybrid recommender systems. Section 3 offers a collection of definitions and concepts associated with this work. Section 4 illustrates the proposed HrecSys framework for designing and building hybrid recommender systems based on two innovative sub-recommenders: a collaborative CF-SVD model and a content-based CB-TFIDF model. Section 5 presents the results of the experimentations with HrecSys over a real-world dataset. Finally, we conclude in section 6.

2. RELATED WORK

To benefit from personalized systems in E-learning platforms, including and developing recommender systems becomes an important research field. Therefore, many kinds of research have been conducted in this field. Some of these works are as follows: The models of past educational recommender systems are based on conventional information retrieval and information filtering. Rodríguez et al. [20] implemented a basic hybrid recommendation approach that

used three methods: content-based, collaborative-based, and knowledge-based, in addition to metadata information about educational resources and students' profiles constructed from their learning styles. Ansari et al. [15] developed a basic hybrid recommender system. They included the context data and implicit feedback to better analyze the students' characteristics. Chen et al. [21] proposed recommender systems for the e-learning field based on hybrid filtering. The approach used two hybrid recommendation systems; the first combines content filtering and collaborative filtering, while the second combines collaborative filtering and the k-neighbour algorithm. In recent years and due to the Machine Learning advancement, the attention is slowly moving toward including deep learning techniques in recommender systems, especially in the E-learning field. Wang et al. [22] implemented a deep learning recommendation framework specified for the E-learning field, in which the model is trained using conventional K-Nearest Neighbor (KNN). The system collected various students' behaviors and grouped them into similar groups based on an ensemble of information such as grades and preferences, then applied different deep learning models to similar groups to make recommendations. In the meantime, they used a conventional three-parameter logistic model to give the parameters related to items. Chang et al. [23] presented a popular-based recommender system using machine learning techniques. The users are clustered using the unsupervised machine learning algorithm K-MEANS based on their interacted items. Then, the system determines the popular groups based on those clusters, and k nearest neighbour KNN is used to determine the closest group. Finally, the user will get popular item recommendations inside the selected group. Mediani et al. [24] have proposed an e-learning system that combines popularity and collaborative filtering approaches to improve the recommendation accuracy evaluated by applying each approach's recall, precision, and f1-score. They showed that the obtained results exhibit an encouraging performance of their model.

Content-based and collaborative recommendations have often been considered complementary [25]. The hybridization of these two techniques to deal with the shortcomings of each method used alone and take advantage of their strengths has been the subject of several research works.

Geetha et al. [26] is a movie hybrid recommender system that combines collaborative filtering and a content-based approach to build a system that provides more precise recommendations concerning movies. Frolov and Oseledets [27] proposed HybridSVD system used the Singular Value Decomposition to incorporate user and item features within the SVD formula. They have used different similarity metrics considering all pairwise distances and allowing the insertion of side information.

3. BACKGROUND

This section presents a collection of definitions and concepts related to this work. We start by understanding what defines a preference, user profile, and item profile. Then we will explain the technologies used in the implementation.

3.1 Preference

A preference is a ranking formula of a collection of items that the user is interested in. It is formulated as follows:

$$pr(u, i) = \{u, i, w_{u,i}\} \quad (1)$$

where, $pr_{u,i}$ represents the user preference. It is composed of a user $u \in U$ and a preferred item $i \in I$ with a given weight $w_{u,i}$. This last is usually a real number representing the user's interest level regarding the item. The interest weight differs from one approach to another. In a content-based approach, the preference weight generally means the distance between the vector of the user and an item vector. Instead, in collaborative filtering, the preference weight is the rating users give to items.

3.2 User profile

A user profile has different representations based on the adopted hybrid method: content-based filtering or collaborative filtering. In collaborative filtering, a user profile is represented as a vector of his interacted items. However, we represent each user and each item by their embedding in content-based filtering because a sequence of words represents an item. Therefore, the user profile also is defined by a sequence of words from his read texts.

3.3 Application of TF-IDF

We applied TF-IDF to evaluate the importance of resources. TF-IDF [28] is a statistical method used to research information in textual documents. It measures the relevancy of a term to a document. A term has a heavier weight when it appears more frequently in a document. The weight also varies with the frequency of the term in the corpus. TF-IDF concentrates on the relation between terms, documents, and corpus. If a term appears more in a document and appears less in the other documents of the same corpus, it better represents this document [29].

We are interested in evaluating the correlation between users and shared articles for our research. In our case, we study the relationship between users and interacted articles shared on the platform. So we can recommend to this user articles that have not already interacted with them.

The basic formula of TF-IDF is as follows [30]:

$$tf(t, d) = \frac{f_{t,d}}{\sum_k f_{k,d}} \quad (2)$$

$$idf(t, D) = \log \frac{N}{df(t)} \quad (3)$$

$$tf - idf(t, d, D) = tf(t, d) \times idf(t, D) \quad (4)$$

where, t is a term, d is a document in the corpus, D is a collection of documents, N is the number of documents that appear in the corpus, df is the number of documents where the term t appears, $f_{t,d}$ is the number of times the term t occurs in document d .

The Cosine similarity defines user similarity as the cosine value of the angle between two vectors projected into multidimensional space. This measure determines how a document is similar to user documents that this user has liked in the past. The documents are represented by weight vectors, where each weight indicates the association degree between the document and the term. Given two documents d_j, d_k represented by weight vectors, and their similarity is measured by [30]:

$$Sim(d_j, d_k) = \frac{d_j \cdot d_k}{\|d_j\| \cdot \|d_k\|} = \frac{\sum_{i=1}^n w_{i,j} \cdot w_{i,k}}{\sqrt{\sum_{i=1}^n w_{i,j}^2} \cdot \sqrt{\sum_{i=1}^n w_{i,k}^2}} \quad (5)$$

where, $w_{i,j}$ is the weight of the term i in the document j and $w_{i,k}$ is the weight of the term i in the document k .

3.3.1 Example

To explain the basic formulas of TF-IDF, we gave an example of a corpus D composed of three documents $d1, d2$, and $d3$. Each document contains a set of words among:

$$T = \{t1, t2, t3, t4, t5, t6, t7, t8, t9, t10, t11, t12, t13\} \quad (6)$$

$$d1 = \{t2, t3, t4, t8, t11, t12\} \quad (7)$$

$$d2 = \{t1, t2, t5, t6, t12\} \quad (8)$$

$$d3 = \{t1, t6, t7, t9, t10, t6, t13\} \quad (9)$$

The corpus D is formulated as:

$$D = [d1, d2, d3] = [[t2, t3, t4, t8, t11, t12], [t1, t2, t5, t6, t12], [t1, t6, t7, t9, t10, t6, t13]] \quad (10)$$

Now we evaluate the TF-IDF of the term "t2" in the document "d1". According to this example, "t2" appears two times in the corpus D , among which it appears one time in "d1". According to the TF-IDF method of evaluation, by applying Eqns. (2), (3), and (4), we obtain:

$$tf(t2, d1) = \frac{f_{t2,d1}}{\sum_k f_{k,d1}} = \frac{1}{6} \quad (11)$$

$$idf(t2, D) = \log \frac{N}{df(t2)} = \log \frac{3}{2} = 0.4054 \quad (12)$$

$$\begin{aligned} tf - idf(t2, d1, D) &= tf(t2, d1) \times idf(t2, D) \\ &= \frac{1}{6} \times 0.4054 = 0.0675 \end{aligned} \quad (13)$$

After calculating the TF-IDF of all the words, each document is represented by a numerical vector, and the corpus becomes:

$$\begin{aligned} D_transform = & [[0.0, 0.0675, 0.1831, 0.1831, 0.0, 0.0, \\ & 0.0, 0.1831, 0.0, 0.0, 0.1831, 0.0675, 0.0], [0.0810, \\ & 0.0810, 0.0, 0.0, 0.2197, \mathbf{0.0810}, 0.0, 0.0, 0.0, 0.0, \\ & 0.0, 0.0810, 0.0], [0.0579, 0.0, 0.0, 0.0, 0.0, \mathbf{0.1158}, \\ & 0.1569, 0.0, 0.1569, 0.1569, 0.0, 0.0, 0.1569]] \end{aligned} \quad (14)$$

According to this corpus, the word "t6" appears one time in "d2" but two times in "d3". $tf-idf(t6, d2, D) = 0.0810$, and $tf-idf(t6, d3, D) = 0.1158$.

The document $d3$ thus appears as "the most relevant" for the word $t6$.

3.4 Application of SVD

The Singular Value Decomposition SVD is a linear algebra method used as a dimensionality reduction technique in machine learning. SVD is a matrix factorization technique used to reduce the number of matrix features by reducing the

space dimension from N-dimension to K-dimension, where $K < N$. SVD affirms that a matrix A can be factorized as [31]:

$$A = USV^T \quad (15)$$

U and V represent orthogonal matrices with orthonormal eigenvectors chosen from AA^T and $A^T A$, respectively. These last matrices have the same positive eigenvalues. S represents a diagonal matrix having r elements equal to the root of the positive eigenvalues of AA^T or $A^T A$. The diagonal elements are singular values. So the matrix $A_{m,n}$ can be factorized as follows:

$$A_{m \times n} = U_{m \times r} S_{r \times r} V_{r \times n}^T \quad (16)$$

We complete U_r and V_r with $u_{r+1}, u_{r+2}, \dots, u_m$ and $v_{r+1}, v_{r+2}, \dots, v_n$ to obtain U et V square and orthogonal matrices.

$$A_{m \times n} = U_{m \times m} S_{m \times n} V_{n \times n}^T \quad (17)$$

where, $S_{m \times n}$ is a diagonal matrix, which can be non-square. U and V^T are orthonormal matrices, which means the columns of U or rows of V are orthogonal to each other and are unit vectors. One of the orthonormal matrix properties is that its transpose is its inverse. So, if U is an orthonormal matrix, we have $U^T = U^{-1}$ or $U.U^T = U^T.U = I$, where I is the identity matrix.

The decomposition of Eq. (17) is called full SVD, whereas Eq. (16) is called reduced SVD because we have $S_{r \times r}$, a square diagonal matrix with r the rank of matrix A . r is usually less than or equal to the smaller of m and n . The matrices $U_{m \times r}$ and $V_{r \times n}^T$ are non-square.

In the recommender system context, A represents the users-items utility matrix where n is the number of users and m is the number of resources [32]. If matrix A is rank r , the matrices AA^T and $A^T A$ are both rank r . In singular value decomposition, the columns of matrix U and the rows of matrix V^T are the eigenvectors of AA^T and $A^T A$, respectively.

SVD is used to decrease the dimension of the utility matrix by extracting its latent factors. Each user and item are mapped into a latent space with dimension r to better understand the relationship between users and items. If A originally contained users as the rows and resources as its columns, the k -dimensional rows of US contain the latent factors of the users, and the k -dimensional columns of SV^T contain the latent factors of the resources.

4. PROPOSED ARCHITECTURE

Our recommender system recommends pedagogical resources that correspond to the learner's interests. We have adopted a hybrid approach composed of two principal recommender system methods: a content-based approach and a collaborative filtering approach. Also, we used the weighting hybrid strategy to combine these two approaches. The global architecture of our recommender system is composed of two main modules: CB approach and CF approach. Our view is to use the benefits of a hybrid recommender system to fill the weaknesses of individual recommendation approaches. Therefore, our built system used a weighted strategy between content-based and collaborative approaches. Then the system creates the final student recommendation list using the weighted scores of both content-based and collaborative-based.

The architecture concepts and details are explained as follows:

4.1 Content-based module

The content-based approach digs into users' profiles and provides recommendations based on their past behavior in this module. Usually, raw texts (unstructured texts data) are the most important type of content in the context of the pedagogical resources. Thus, our objective consists of using keywords extracted thanks to automatic indexing and which are representative of each item or profile. For that reason, we chose to adopt the TF-IDF technique to improve the accuracy of the content-based recommendation.

The algorithm used in this module is:

Algorithm 1: CB-TFIDF

Input: U : list of users interactions: size n

I : list of items: size m

Output: Recommendation lists of size l .

1. Group all user interactions by user.
 2. For each user
 3. Select all contents that the user has interacted with.
 4. Cluster contents TF-IDF vectors to model user's articles of interest
 5. Weight clusters' relevance to the user by summing each cluster's interactions strength (view, like, comment, follow, and bookmarks).
 6. Returns highly relevant articles vectors for the user
 7. For each user article
 8. Apply cosine similarity to calculate similarity.
 9. Recommend more similar content to user $L1$, with which the user has not yet interacted.
-

4.2 Collaborative filtering module

There have been a few attempts to reduce the number of features in the utility matrix without significantly decreasing final prediction accuracy. This module adapted an SVD matrix factorization technique enhanced to improve recommendations accuracy. The model is created from the fusion of the SVD factors to model user-item latent structures. The algorithm is explained as follows:

Algorithm 2: CF-SVD

Input: U : list of users interactions: size n

I : list of items: size m

Output: Recommendation lists of size l .

1. For each pair of user-item interactions, aggregate different implicit interaction weights $w_{u,i}$ regard an item to get a unified value $W_{u,i}$:

$$W_{u,i} = \sum_{k=1}^n w_{u,k} \quad (18)$$

where, $k \in K$ is the number of the various possible interactions of the system.

2. Build the user-item utility matrix y_{ui} , that contains all the users $U = \{u_1, u_2, u_m\}$ arranged as rows, and items $I = \{i_1, i_2, i_n\}$ arranged as columns. The constructed matrix is very sparse, where a 0 is assigned to non-interacted items.
3. Decompose the utility matrix into the product of three matrices: two orthonormal matrices U and V , and a diagonal matrix S using Eq. (16).

4. Reconstruct an approximation of the original matrix by multiplying its factors. The resulting matrix is not sparse anymore. It generated predictions for items the user has not yet interacted with, which we will exploit for recommendations.
5. Normalize the obtained matrix by using the followed formula:

$$x_{norm} = \frac{x - x_{mean}}{x_{max} - x_{min}} \in [0,1] \quad (19)$$

6. For each user, generate a candidate ranked list of items L2 close to the user profile, based on the predictions provided by the SVD decomposition.

4.3 Hybrid module

We mentioned that our system utilizes two recommender approaches, CB-TFIDF and CF-SVD. We use a weighted strategy to combine the two main modules. Each module generates a candidate list of items (L1 for CB-TFIDF and L2 for CF-SVD) on the system that might interest the user. Each list represents the student's primary profile $p(u)$. The content-based approach is responsible for finding similar not-popular items of $p(u)$ based on the content by using item embedding and distance measures. Additionally, the system calls for collaborative filtering to get what other students prefer. Then a hybrid weighted strategy is used to collect the scores of both CB-TFIDF and CF-SVD. Therefore, the student got a hybrid recommended list L representing the combined lists L1 and L2 recommendations. The score of L is calculated as follows:

$$L.score = P1 \times L1.score + P2 \times L2.score \quad (20)$$

where, $P1$ and $P2$ are two numerical parameters, in our case, as the CF model is much more accurate than the CB model, the weight of $P1$ for the CF model is superior to $P2$ of the CB model.

The Figure 1 wraps all the explained steps of each used recommender system approach.

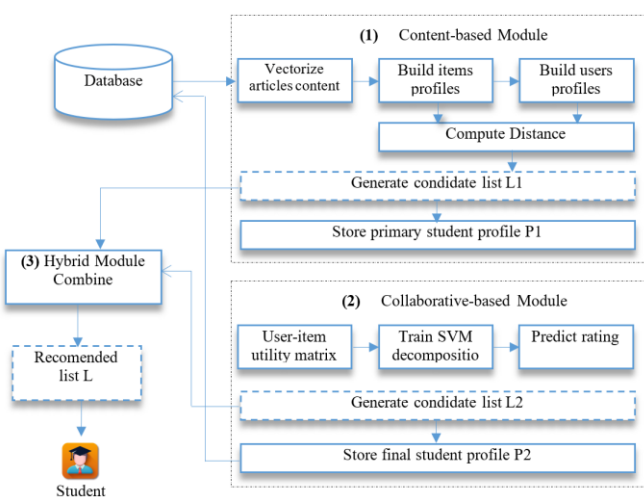


Figure 1. Our system detailed architecture

5. EXPERIMENTAL RESULTS

This section covers the assessments used to evaluate the

proposed method. The dataset used in the experiments, the numerous evaluation metrics of the used models, the competing works on which we based our research, the system configuration, and the parameter settings and performance of all our employed approaches are also covered.

5.1 Data collection and preprocessing

Users can utilize our recommender system to find pedagogical resources. To represent these resources, we have chosen articles. As a result, we trained and tested our recommender system using the popular available Articles dataset (DeskDrop). The following is a description of the dataset and how it was prepared:

5.1.1 Definition

The dataset comprises two data files (Table 1): Articles sharing and users interactions, both of which are twelve-month samples (March 2016 - February 2017) from the DeskDrop internal communication platform. These allow the companies' employees to exchange relevant content with their colleagues and collaborate around them. About 73k registered user interactions on more than 3k public articles shared on the platform make up the collection.

Table 1. Data description

Name	Description	Number of samples
shared_articles	It contains public articles shared in the Internal Communication Platform DeskDrop.	3128
users_interactions	It contains logged users' interactions on the articles shared on the platform	72312

5.1.2 Dataset files

The dataset contains two files:

- Shared Articles: This file contains information about articles shared on the platform. Each article includes information on the date of sharing (timestamp), the original (URL, title, plain text content), the language in which the article was written (lang) (Portuguese-pt or English-en), and the user who shared the article (author).

- User Interaction: This file contains user log information about the platform's shared article interaction. The most important feature is (eventType), which covers all the different user interactions (view, like, comment, follow, bookmark).

5.1.3 Dataset features

The dataset contains these features:

- (1) Item attributes: The original URL of the article, the title, and the plain text of the content, which are available in two languages (English and Portuguese).

- (2) Contextual information: The context of the user's visit, such as date/time, the client (mobile native app/browser), geographic location, etc.

- (3) Logged users: All users need to log into the platform to keep track of their preferences over the long term.

- (4) Rich implicit feedback: Various interaction types are logged, making it possible to infer the level of interest of the user in the article (comments, likes, views, follow, bookmark).

5.1.4 Dataset pre-processing

The dataset contains several implicit user interactions, such as views and comments. However, to pass these interactions to machine learning algorithms, each type of interaction is assigned appropriate weights depending on the level of interaction (Table 2) [33].

Table 2. Associated interaction weight

Interaction	Weight
View	1
Like	2
Comment	2,5
Follow	3
Bookmark	4

Data engineering is the second stage of preprocessing, which includes removing duplicates and removing unnecessary features. We also used pre-trained English word2vec word embedding. So instead of deleting non-English articles, we used the translation API to unify languages. It helps to get more data for better model training.

5.2 Evaluation metrics

The study provides different evaluation indicators carried out in this area to evaluate the quality and productivity of the built recommender system. The most widely used indicators of the accuracy of recommendations are Precision@k and Recall@k. The top-N items are the most important to the user in the RS context. So, it makes sense to calculate precision and recall metrics in the first N elements rather than in all elements. Each user's number of elements evaluated is usually smaller than those available in the entire dataset. The number of relevant elements in the test set may also be smaller than the whole dataset. Therefore, unlike the information retrieval field, in the recommender systems field, precision and recall depend on the number of items assessed per user [34]. RS measures have two of the most important terms: recommended and relevant. The term true positive (TP) indicates the relevant element recommended to the user. True Negatives (TN) are irrelevant items that are not recommended to users, False Negatives (FN) are the relevant items that are recommended to users, and False Positives (FP) are irrelevant items that are recommended to users. Recommended items are generated by the model when the relevant items are known in the dataset [34].

(1) Precision@k: Precision is the proportion of user-recommended and relevant items (TP) out of the total number of recommended items (TP and FP) [34]:

$$Precision@K = \frac{Recommended \cap Relevant}{Recommended} = \frac{TP}{TP + FP} \quad (21)$$

(2) Recall@k: Recall shows the ratio of relevant items recommended to the user (TP) from the total number of relevant items for the user (TP and FN) [34]:

$$Recall@K = \frac{Recommended \cap Relevant}{Relevant} = \frac{TP}{TP + FN} \quad (22)$$

(3) F1-score: The F1-score is a harmonic means between recall and precision [34]:

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (23)$$

(4) Accuracy: Accuracy is the percentage that the model correctly predicted.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} = \frac{TP}{TP + TN + FP + FN} \quad (24)$$

Our recommending system evaluated the performance of the approach using the cross-validation approach called the hold-out strategy. According to this strategy, random samples from the dataset are kept as training data, and the remaining data are test data. The data divided into training and testing includes various suggestions (80% training, 20% test). For each user, ranking all items in the dataset is time-consuming. Therefore, we use a popular strategy of randomly sampling the top K items (100 items in our case) with which the user has not yet interacted. It then ranks the missing items in these top-K item lists and computes an accuracy metric from the recommended ranking list for that user and interaction item. Our recommender system uses Precision@10, Recall@10, F1-score, and Accuracy to measure the accuracy of recommendations. However, in our content-based approach, we use cosine similarity to measure the distance between the user and the item TF-IDF embeddings. We also use Precision@10, Recall@10, F1-score, and Accuracy to measure collaborative filtering and hybrid recommender performance.

5.3 System configuration

All of our testings are done on a DELL Intel (CORE i5) with an Intel(R) Core(TM) processor i5-4310U CPU @ 2.00 GHz 2.60 GHz and 4.00 GB RAM.

5.4 Parameter settings

Our recommender system is based on different approaches included in the system. We have used different configuration settings in our system implementation.

For the content-based approach, we used a TF-IDF model. This model used the TfidfVectorizer() class from *sklearn.feature_extraction.text* library to calculate and vectorize each article's Tf-IDF scores. TfidfVectorizer will convert the title and the plain text of the sharing article (text columns) into numerical.

For the collaborative filtering approach, we used the SVD latent factor models. We chose a SciPy implementation of SVD to implement our model. Latent factor models compress the utility matrix into a low-dimensional matrix in terms of latent factors. Another important decision is the number of factors to factor the utility matrix. If the number of factors is higher, factorization in the utility matrix reconstructions is more precise. Reducing the number of factors increases the generalization of the model. Therefore, if the model is allowed to save many details of the utility matrix, it may not generalize well for untrained data.

5.5 Competing approaches

To validate our approach, we have implemented all the previously mentioned recommendation approaches, content-based, collaborative-based, and hybrid-based. We implemented each one separately and compared all of them. We have used recall, precision, F-score, and accuracy as

recommendation measures.

5.6 Results and discussion

This section illustrates the results of various approaches and techniques used in the research. It also covers each approach's performance in detail and compares the final recommendations accuracy of each one.

5.6.1 Content-based model

After building the matrix for TF-IDF features, we used cosine similarity to measure the distance between users and items vectors. A sampled user presented in (Table 3 and Table 4) shows that the system discovered that the user -1443636648652872475 has a recall@10 of 0.4017 and a precision@10 of 0.8245.

Our content-based approach scores a global recall of (0.26). The placeholder (26%) of interacted articles in the test set were ranked among the top-10 items by this model. It also scores a global precision of (0.12), which means (12%) of recommended articles among the top-10 items are relevant to the user. Table 3 and Table 4 show some users' details recall@10 and precision@10, respectively.

Table 3. CB-TFIDF model recall for some users

PersonId	Hits@10	Interacted items	Recall@10
-1443636648652872475	47	117	0.4017
-3596626804281480007	25	80	0.2875
-1032019229384696495	33	130	0.2538

Table 4. CB-TFIDF model precision for some users

PersonId	Hits@10	Recommended	Precision@10
-1443636648652872475	47	57	0.8245
-3596626804281480007	25	35	0.6969
-1032019229384696495	33	43	0.7674

5.6.2 Collaborative filtering model

Our collaborative filtering has used the conventional matrix factorization technique of singular value decomposition. Our approach scored a global recall of (0.47) for the top-10 items. It means that 47% of interacted articles in the test set were ranked among the top-10 items by this model. It also achieves a global precision of (0.18) for the top-10 items, which means (18%) of recommended articles are relevant to the user. Table 5 and Table 6 show some users' details recall@10 and precision@10.

Table 5. CF-SVD model recall for some users

PersonId	Hits@10	Interacted items	Recall@10
-1443636648652872475	51	117	0.4358
-3596626804281480007	34	80	0.4250
-2979881261169775358	48	88	0.5454

Table 6. CF-SVD model precision for some users

PersonId	Hits@10	Recommended	Precision@10
-1443636648652872475	51	61	0.8360
-3596626804281480007	34	44	0.7727
-2979881261169775358	48	57	0.8421

5.6.3 Hybrid model

Our hybrid system scores a significance recall of (0.48) and a global precision of (0.19) for the top-10 items. The hybrid approach surpasses each separated one because it uses the weighted scores of both content-based and collaborative-based models. Table 7 and Table 8 show some users' details recall@10 and precision@10.

Table 7. HrecSys model recall for some users

PersonId	Hits@10	Interacted items	Recall@10
-1443636648652872475	51	117	0.4358
-3596626804281480007	35	80	0.4375
-2979881261169775358	50	88	0.5681

Table 8. HrecSys model precision for some users

PersonId	Hits@10	Recommended	Precision@10
-1443636648652872475	51	61	0.8360
-3596626804281480007	35	45	0.7777
-2979881261169775358	50	59	0.8474

From Figures 2, 3, 4, and 5, which represent the metrics scores of the proposed model, we can conclude that our proposal outperforms the separated approaches. It is the best along the four metrics used. Our hybrid model was superior in predicting the recommended resources. It reached an accuracy@5 of 92%, accuracy@5 of 88%, and accuracy@5 of 80%.

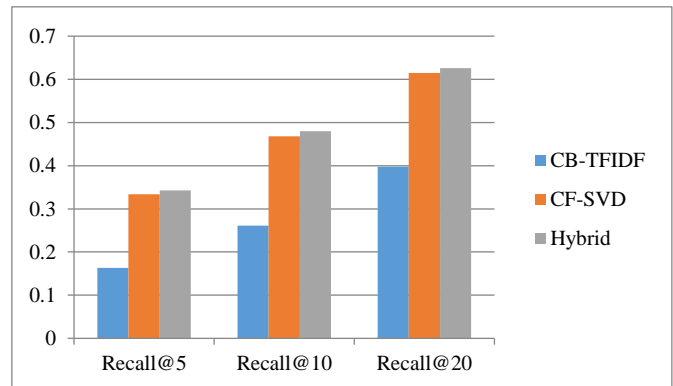


Figure 2. Recommendations Recall for CB-TFIDF, CF-SVD, and HrecSys

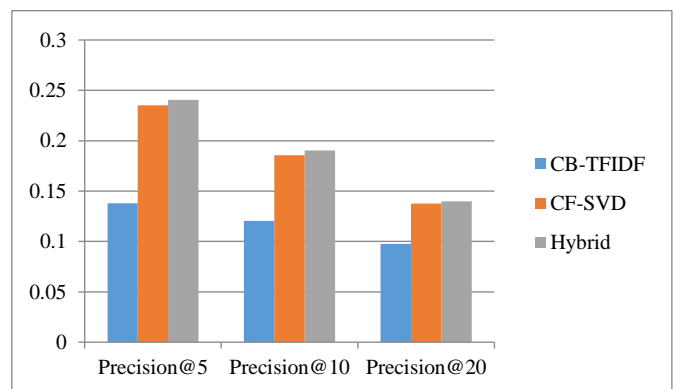


Figure 3. Recommendations precision for CB-TFIDF, CF-SVD, and HrecSys

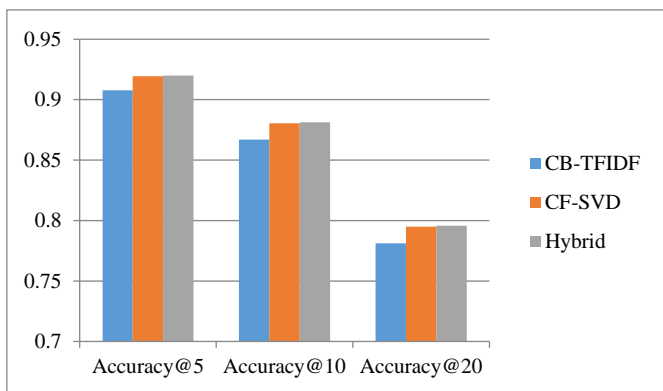


Figure 4. Recommendations accuracy for CB-TFIDF, CF-SVD, and HrecSys

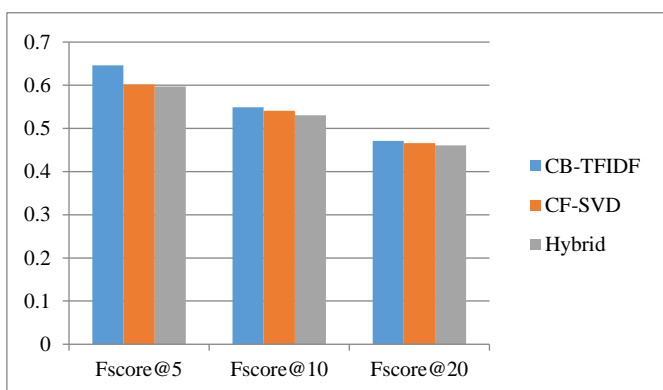


Figure 5. Recommendations Fscore for CB-TFIDF, CF-SVD, and HrecSys

6. CONCLUSIONS

Recommender systems have become, like the search engine, an essential tool for any website and online platform. E-learning starts to include and improve recommendation systems by providing learners with personal and efficient pedagogical resources to help them have a better learning experience. We applied a couple of machine learning technologies to improve recommender systems of pedagogical resources. In this work, we used the hybridization of two recommender system approaches, content-based and collaborative filtering. We used Tf-IDF technology to let the system better understand the meaning of words of pedagogical resources in the content-based approach. And we used an SVD matrix factorization to predict users' ratings better in collaborative filtering. Our system used a hybrid weighted strategy to coordinate the hybridization approaches. It is a weighted score of content-based and collaborative filtering. Our system shows encouraging results in the experiment tests, where it outperforms some existing models on real-world datasets.

We believe that deep learning-based models will play a crucial role in e-learning systems or intelligent tutoring systems in the future. We believe our work can be improved upon to produce intriguing results. In future works, we could apply a couple of deep learning technologies to improve the recommender systems of pedagogical resources. We could use word embeddings technology (Wordtovec model, Glove model, etc.) to let the system better understand the semantics and meaning of words of pedagogical resources in the content-

based approach. On the other hand, we could simulate low-rank matrix factorization using a deep neural network in collaborative filtering to reduce the rating-matrix dimensionality. We could also prevent the user cold-start problem and provide new students with recommendations in an e-learning platform by using a popularity-based approach as a baseline to provide new users with the most trending resources on the system.

REFERENCES

- [1] Ricci, F., Rokach, L., Shapira, B. (2011). Introduction to recommender systems handbook. In *Recommender Systems Handbook*, 1-35. https://doi.org/10.1007/978-0-387-85820-3_1
- [2] Burke, R. (2007). Hybrid web recommender systems. In *The Adaptive Web. Lecture Notes in Computer Science*, Berlin, 4321: 377-408. https://doi.org/10.1007/978-3-540-72079-9_12
- [3] Zhang, S., Yao, L., Sun, A., Tay, Y. (2019). Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)*, 52(1): 1-38. <https://doi.org/10.1145/3285029>
- [4] Ko, H., Lee, S., Park, Y. Choi, A. (2022). A survey of recommendation systems: Recommendation models, techniques, and application fields. *Electronics*, 11(1): 141. <https://doi.org/10.3390/electronics11010141>
- [5] Bressan, M., Leucci, S., Panconesi, A., Raghavan, P., Terolli, E. (2016). The limits of popularity-based recommendations, and the role of social ties. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco California USA, 13-17-Aug, pp. 745-754. <https://doi.org/10.1145/2939672.2939797>
- [6] Zhou, R., Khemmarat, S., Gao, L. (2010). The impact of YouTube recommendation system on video views. *Proceedings of the ACM SIGCOMM Internet Measurement Conference IMC*, pp. 404-410. <https://doi.org/10.1145/1879141.1879193>
- [7] Yang, J. (2016). Effects of popularity-based news recommendations ("most-viewed") on users' exposure to online news. *Media Psychology*, 19(2): 243-271. <https://doi.org/10.1080/15213269.2015.1006333>
- [8] Bobadilla, J., Ortega, F., Hernando, A., Guti'erez, A. (2013). Recommender systems survey. *Knowledge-Based Systems*, 46: 109-132. <https://doi.org/10.1016/j.knosys.2013.03.012>
- [9] Mooney, R.J., Roy, L. (2000). Content-based book recommending using learning for text categorization, *Proceedings of the ACM International Conference on Digital Libraries*, pp. 195-204. <https://doi.org/10.1145/336597.336662>
- [10] AL-Ghuribi, S.M., Noah, S.A. (2021). A comprehensive overview of recommender system and sentiment analysis. *ArXiv*. <https://doi.org/10.48550/ARXIV.2109.08794>
- [11] Linden, G., Smith, B., York, J. (2000). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1): 76-80. <https://doi.org/10.1109/MIC.2003.1167344>
- [12] Adomavicius, G., Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17:

- 734-749. <https://doi.org/10.1109/TKDE.2005.99>
- [13] Burke, R. (2002). Hybrid recommender systems: Survey and experiments, *User Model. User-Adapted Interact*, 12(4): 331-370. <https://doi.org/10.1023/A:1021240730564>
- [14] Dietmar, J., Zanker, M., Felfernig, A., Friedrich, G. (2010). Hybrid recommendation approaches. In *Recommender Systems: An introduction*. Cambridge University Press, pp. 124-142. <https://doi.org/10.1017/CBO9780511763113>
- [15] Ansari, M.H., Moradi, M., NikRah, O., Kambakhsh, K.M. (2016). CodERS: A hybrid recommender system for an E-learning system. In *2016 2nd International Conference of Signal Processing and Intelligent Systems (ICSPIS)*, pp. 1-5. <https://doi.org/10.1109/ICSPIS.2016.7869884>
- [16] Claypool, M., Gokhale, A., Miranda, T. (1999). Combining content-based and collaborative filters in an online newspaper. In *Proceedings of the SIGIR-99 Workshop on Recommender Systems: Algorithms and Evaluation*.
- [17] Cotter, P., Smyth, B. (2000). PTV: Intelligent personalized TV guides. In *Twelfth Conference on Innovative Applications of Artificial Intelligence*, pp. 957-964.
- [18] Wang, D., Liang, Y., Xu, D., Feng, X., Guan, R. (2018). A content-based recommender system for computer science publications. *Knowledge-Based Systems*, 157: 1-9. <https://doi.org/10.1016/j.knosys.2018.05.001>
- [19] Guan, X., Li, C., Guan, Y. (2017). Matrix factorization with rating completion: An enhanced SVD model for collaborative filtering recommender systems. *IEEE Access*, 5: 27668-27678. <https://doi.org/10.1109/ACCESS.2017.2772226>
- [20] Rodríguez, P.A., Ovalle, D.A., Duque, N.D. (2015). A student-centered hybrid recommender system to provide relevant learning objects from repositories. In *International Conference on Learning and Collaboration Technologies*, pp. 291-300. https://doi.org/10.1007/978-3-319-20609-7_28
- [21] Chen, W., Niu, Z., Zhao, X., Li, Y. (2014). A hybrid recommendation algorithm adapted in E-learning environments. *World Wide Web*, 17(2): 271-284. <https://doi.org/10.1007/s11280-012-0187-z>
- [22] Wang, X., Zhang, Y., Yu, S., Liu, X., Yuan, Y., Wang, F.Y. (2017). E-learning recommendation framework based on deep learning. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 455-460. <https://doi.org/10.1109/SMC.2017.8122647>
- [23] Chang, J.H., Lai, C.F., Huang, Y.M, Chao, H.C. (2010). 3PRS: A personalized popular program recommendation system for digital TV for P2P social networks, *Multimedia Tools and Applications*, 47(1): 31-48. <https://doi.org/10.1007/s11042-009-0405-6>
- [24] Mediani, Y., Gharzouli, M., Mediani, C. (2022). A hybrid recommender system for pedagogical resources. *The International Conference on Digital Technologies and Applications, Morocco, 29-30 January*. In *Proceedings Digital Technologies and Applications*, Springer, pp. 361-371. https://doi.org/10.1007/978-3-031-02447-4_38
- [25] Balabanović, M., Shoham, Y. (1997). Fab: Content-based, collaborative recommendation. *Communications of the ACM*, 40(3): 66-72. <https://doi.org/10.1145/245108.245124>
- [26] Geetha, G., Safa, M., Fancy, C., Saranya, D. (2018). A hybrid approach using collaborative filtering and content based filtering for recommender system. *Journal of Physics: Conference Series*, 1000: 012101. <https://doi.org/10.1088/1742-6596/1000/1/012101>
- [27] Frolov, E., Oseledets, I. (2019). HybridSVD: When collaborative information is not enough. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pp. 331-339. <https://doi.org/10.1145/3298689.3347055>
- [28] Jones, K.S. (1972). A statistical interpretation of term specificity and its application. In *retrieval. Journal of Documentation*, 28(1): 11-21.
- [29] Muthurasu, N., Rengaraj, N., Mohan, K.C. (2019). Movie recommendation system using term frequency-inverse document frequency and cosine similarity method. *International Journal of Recent Technology and Engineering*, 7: 86-90.
- [30] Philip, S., Shola, P., Ovyte, A. (2014). Application of content-based approach in research paper recommendation system for a digital library. *International Journal of Advanced Computer Science and Applications*, 5(10). <https://doi.org/10.14569/IJACSA.2014.051006>
- [31] Zhou, X., He, J., Huang, G., Zhang, Y. (2015). SVD-based incremental approaches for recommender systems. *Journal of Computer and System Sciences*, 81(4): 717-733. <https://doi.org/10.1016/j.jcss.2014.11.016>
- [32] Yuan, X., Han, L., Qian, S., Xu, G., Yan, H. (2019). Singular value decomposition based recommendation using imputed data. *Knowledge-Based Systems*, 163: 485-494. <https://doi.org/10.1016/j.knosys.2018.09.011>
- [33] Mediani, C., Abel, M.H., Djoudi, M. (2015). Towards a recommendation system for the learner from a semantic model of knowledge in a collaborative environment. In *IFIP International Conference on Computer Science and its Applications*, pp. 315-327. https://doi.org/10.1007/978-3-319-19578-0_26
- [34] Cremonesi, P., Turrin, R., Lentini, E., Matteucci, M. (2008). An evaluation methodology for collaborative recommender systems. In *2008 International Conference on Automated Solutions for Cross Media Content and Multi-Channel Distribution*, pp. 224-231. <https://doi.org/10.1109/AXMEDIS.2008.13>