

Research Article

Dealing with Pure New User Cold-Start Problem in Recommendation System Based on Linked Open Data and Social Network Features

Usha Yadav¹ , Neelam Duhan,² and Komal Kumar Bhatia³

¹National Institute of Fashion Technology, Jodhpur, India

²J. C. Bose University of Science & Technology, YMCA, Faridabad, India

³J. C. Bose University of Science & Technology, YMCA, Faridabad, India

Correspondence should be addressed to Usha Yadav; usha.yadav.912@gmail.com

Received 26 September 2019; Revised 19 May 2020; Accepted 25 May 2020; Published 23 June 2020

Academic Editor: Filippo Sciarrone

Copyright © 2020 Usha Yadav et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Preferring accuracy over computation time or vice versa is very challenging in the context of recommendation systems, which encourages many researchers to opt for hybrid recommendation systems. Currently, researchers are trying hard to produce correct and accurate recommendations by suggesting the use of ontology, but the lack of techniques renders to take its full advantage. One of the major issues in recommender systems bothering many researchers is pure new user cold-start problem which arises due to the absence of information in the system about the new user. Linked Open Data (LOD) initiative sets standards for interoperability among cross domains and has gathered enormous amount of data over the past years, which provides various ways by which recommender system's performance can be improved by enriching user's profile with relevant features. This research work focuses on solving pure new user cold-start problem by building user's profile based on LOD, collaborative features, and social network-based features. Here, a new approach is devised to compute item similarity based on ontology, thus predicting the rating of nonrated item. A modified method to calculate user's similarity based on collaborative features to deal with other issues such as accuracy and computation time is also proposed. The empirical results and comparative analysis of the proposed hybrid recommendation system dictate its better performance specifically for providing solution to pure new user cold-start problem.

1. Introduction

Fetching desired information from websites or apps containing enormous data such as items, videos, pictures, and text is a very challenging and time-consuming task. Research domains such as information retrieval and information filtering systems are highly inspired with the advancement done in artificial intelligence approaches. Recommender system (RS) is a competent tool which assists users by providing a ranked list of items as per their requirements or preferences without being explicitly searching in the system. This system has proved to be an important tool to recommend items, thereby personalizing the applications for various domains such as tourism, marketing, movies, songs, hotels and restaurants, news, and forecasting theories.

There are many famous recommendation systems provided by top e-commerce applications such as *Flipkart*, *Amazon Prime Videos*, and *MakeMyTrip*. The main motive of all recommendation systems is to provide the most appropriate items to the right user at the right time. Vast research studies are going in this field, and many different approaches are proposed which take benefit of different types of data and analyzing techniques. There are various issues while designing an appropriate recommendation system such as scalability, high computation, and diversity. But among all, there is one important issue that has majorly gained the attention of researchers is the *Cold-Start Problem*, which arises at the time of registration of new user or adding up new resource or item in the system. Obviously, there would be no information about user's interest or his rating for any particular item in the

system, and recommending appropriate item at that time to the new user is very challenging.

The quality of the recommender system degrades when there is insufficient information or no ratings at all are available [1]. Although cold-start problem is a very famous and potential issue in the recommender systems, various research studies have been done to resolve this problem. Cold-start problem basically split into two categories, namely, *New User Cold-Start Problem* and *New Item Cold-Start Problem*. New user cold-start problem refers to the lack of information about user's interest or very less ratings provided by this user for any particular item in the system. *Pure New User Cold-Start Problem* refers to the problem when no rating at all is provided by the user in the system. With the increasing e-commerce platforms, huge numbers of new users signing every day or less-active users in almost every application create a serious issue for the recommendation systems [2]. Another major issue is new item cold-start problem which refers to the newly added item in any particular system which has very less or no rating provided by the user, so in this scenario analyzing the item and referring it to the user can be a tedious task [3]. New item cold-start problem is also called as the early-rater problem in various literature studies [4].

In recent literature, many hybrid approaches have been proposed to overcome the problem of new user cold start such as cross-domain collaborative filtering using matrix factorization models [5], learning latent factor representation for videos based on modeling the emotional connection between user and item [6], enhanced content-based algorithm using social networking [7], combining social sub-community division and ontology decision model [8], and using social network textual information to model user interest and item [9].

The Linked Open Data (LOD) cloud is an enormous set of RDF statements interconnected together forming a cross-domain ontology graph and wrapping many domains, such as companies, people, geographical locations, movies, music, and books. DBpedia, one of the largest LOD, is known to be the "typical entry point" to these data [10]. The RDF mapping of Wikipedia is commonly considered as the *nucleus* of the emerging *Web of Data*. As DBpedia sets a standard to define properties and classes representing different domain and providing enormous amount of machine-readable data, a major research is going on to investigate how recommender systems can be made beneficial with this overabundance of data [5, 11] and how Linked Data about items can be used for collaborative filtering algorithm [12].

But there is still lack of features on which the similarity between the users is calculated. Similarity measures should not be only confined to the rating given by the users for particular items or just comparing their basic demographic information such as age and location. There is a severe requirement to analyze more and different features which could describe users well enough depending on various domains. For example, let us consider the clothing domain; two users with similar height and weight are most probable to like similar dresses. Therefore, it is highly required to build up user's profile based on various numbers of features

as per the domain for which that recommender system needs to be developed. The similarity between the users should be calculated to improve the performance of the recommendation system. The main problem lies in the standardization of the features which could successfully describe a user's attributes and provide related data. These data can further be used in representing user's profile in various domains. This work focuses on resolving the cold-start problem by finding the similarity between users based on their social, collaborative, and Linked Open Data features.

The paper is organized into the following sections. Section 2 describes the benefit of using domain-related ontology, semantic web, and Linked Open Data in recommender systems. Section 3 presents a survey of research work done in various types of recommendation systems. Section 4 describes in detail the proposed framework for the recommendation system. Section 5 describes the detailed description of the features adopted to build user profile. Section 6 provides the comparative analysis of existing work done and proposed work. Section 7 shows the experimental results and Section 8 outlines the conclusion and future extensions of the proposed work.

2. Background Theories

2.1. Ontology in the Recommender System. The main purpose of using ontologies is to model the information related to any domain at the semantic level [13]. Ontologies have different definitions in different fields but specifically in context with computer science, it was provided by Gruber [14] and was later refined by Guarino et al. [13]. The notion of ontology is originally defined by Gruber [14] as an "explicit specification of a conceptualization."

Ontology is termed as a graph $O = (C, \mathcal{R}, \mathcal{L})$ which is a directed labeled graph. $C = \{c_1, \dots, c_n\}$ is a set of classes or properties representing the concepts of an ontology where n is the number of nodes defining concepts. $\mathcal{R} = \{r_1, \dots, r_m\}$ denotes the set of direct edges representing all the relationships between the concepts in ontology O , where m is the number of edges defining relationship. $rk \in \mathcal{R}$ represents a directed relationship between two adjacent concepts $c_i, c_j \in C$, i.e., $rk = (c_i, c_j)$. $\mathcal{L} = \{\ell_1, \dots, \ell_m\}$ denotes a set of labels that show the name of each concept in graph node.

Adopting the ontologies in the recommender system had successfully overcome various shortcomings mentioned in [15, 16]. Also, the fuzzy ontologies had been greatly used to improve the accuracy of recommender systems. In [17], domain ontologies were used to predict the most appropriate item as per the user's preferences. Most of the research work frames the semantic recommendation approach combining item-based collaborative filtering and item-based semantic similarity techniques.

2.2. Semantic Web and Linked Open Data (LOD). The goal of semantic web, according to its original vision [18], was all knowledge available on the web in a machine-readable format so that machines would be able to process the vast amount of information. This would require huge effort by

agreeing on a common framework for embedding information and data. Using various knowledge representation languages such as RDF or OWL and protocols such as URI would allow interoperability, thus enabling the data to be shared and reused across many applications, platforms, and communities. Some commendable progress towards this vision has been attained with the recent growth in the Linked Open Data (LOD) initiative [19], and the objective is to link data among various isolated applications, highlight the importance to bring out data publicly for other applications to use, and link the data with each other using standard schema.

Collaborative efforts have made the LOD initiative very successful. As per the recent statistics, “150 billion of RDF triples and almost 10,000 linked datasets are now available in the so-called LOD cloud, a huge set of interconnected semantic datasets whose *nucleus* is commonly represented by DBpedia” [10]. Data from DBpedia can be freely and easily extracted using defined properties as mentioned in its ontology via SPARQL query language. This valuable information or data presented in linked knowledge base can be effectively used in the area of recommendation systems [11], so as to overcome some of its major issues. As LOD contain topical domain’s ontology containing attributes which described that particular domain well enough, so it can easily tackle various issues related to *Content-based Recommender Systems* [20]. For instance, sometimes there are very limited features available to describe the items which need to be recommended, and hence, their content analysis cannot be done efficiently, thus arising the problem of *limited content analysis*. Also in case of user-based collaborative filtering, LOD provides standard schema and data to represent various features related to user domain, which will also enhance the analysis of users while recommending the most appropriate items to them.

3. Related Work

In previous studies, several methods have been developed for recommendation systems. This section explains the prevalent research work related to the proposed work. The typical methods of recommender systems are categorized into collaborative filtering, content-based filtering, and hybrid approach [21]. By adopting the standards, our proposed framework is classified as *hybrid recommender system*.

Researchers improve the hybrid recommendation system for movie domain based on demographic and collaborative filtering-based approach. Their strategy categorizes the genres of movies based on demographic attributes, e.g., user age (child, teenager, or adult), student (yes or no), have children (yes or no), and gender (female or male) [22].

Authors proposed the solution for cold-start problem by exploiting blog textual data and labeling them as per the user’s opinion and then constructed user-item rating matrix for collaborative filtering and improving recommendations [23]. The study of Liu et al. [24] improves the study of Ahn [1] by presenting the concept called Proximity-Significance-Singularity which improves the disadvantages of Pearson correlation coefficient and cosine similarity [25] to improve new user cold-start problem.

Researchers developed recommendation system to offer best suitable Cloud Platform as a Service (PaaS) for application developer algorithm based on ontology. Their experimental analysis shows that work deals with the problem of scalability [26].

Researchers developed a recommender system based on semantic web technique and modeled all the information based on graph language which is Ontology Web Language 2 (OWL 2). This recommender system comes under a hybrid category combining various approaches like combined content-based, context-aware, and CF. Experimental evaluation was done using MovieLens dataset, and results are shown using *F1* measures such as precision and recall [27].

A system for recommending e-learning resources to the learners using an ontology and sequential pattern mining (SPM) was proposed which comes under hybrid knowledge-based recommender system. Authors use ontology to model various learning methods and learning resources and used SPM to find more about user’s sequential learning patterns (Tarus, Niu, and Yousif, 2017). A complete framework was put forward using various web mining techniques and ontologies based on different domains to overcome major issues like cold-start problems, sparsity, and scalability. MovieLens dataset was evaluated using various precision metrics [15].

In many of the latest literature studies, authors have exploited DBpedia majorly to define or modify various similarity measures using its properties gathered from LOD [28, 29]. Social network platform like Facebook is used to gather user’s music preferences and DBpedia is used for calculating similarities between various music items and building personalized playlist for users [30]. *Limited content analysis* is the core issue where Linked Open Data have significantly played a major role and many researchers are taking advantage of using it. Researchers developed an application named *TasteWeights*; it is a kind of recommender system in which user’s preferences for music genre are extracted from Facebook and then DBpedia is exploited using SPARQL query endpoint to find all the music played by new artist belonging to the same genre which the active user liked and then recommending the same to other users [31].

Various matrix factorization models were evaluated for collaborative filtering for cross domain by using the Linked Open Data, which acts as a connector to analyze the items liked by users in different domains. The metadata extracted from the linked open data helps in generating the relationship between the items belonging to different domains. But this approach has some limitations, as it can rely only on those domains who share information with other domains. If the source and the target domain are closed domain, it would not be possible to share information and hence semantic linking between the items would not be exposed [5].

Systematic literature review of all the research works published between 2011 and 2017 had been done on mitigating the cold-start problem using social network and collaborative filtering for providing recommendations. According to the research done over the period, it is found that the number of research studies increases which focused

on mitigating the cold-start problem using social network and these are published in reputed journals or conferences having high citation [32].

In this work, a hybrid recommendation system framework is proposed for the motive of solving pure new user cold-start problem, using proposed domain-based ontologies algorithm, explicit user-item rating matrix, and clustering of similar items. The sparsity issues are resolved by suggesting algorithm for rating prediction. Also for efficiently finding similarity between users and comparing them with new user, *User Profile Generation Module* is proposed to make use of features directly extracted from the LOD cloud, social network graph, and collaborative features. All these are the distinguishing aspects of this work.

4. Proposed Framework

In the proposed framework, as shown in Figure 1, there are various modules involved to figure out the possible and accurate solution for pure new user cold-start problem. The proposed architecture is briefly explained as follows:

- (i) First of all, system is provided with user-item matrix in which the rating given to each item by the users is specified. So, to find the similarity between items, item clusters need to be generated using fuzzy C means clustering.
- (ii) To cluster items based on similarity, the pairwise similarity between the items is calculated. The average of ontology-based similarity and item-based similarity is calculated as the overall similarity. New algorithm is proposed to calculate the item similarity based on ontologies.
- (iii) Once the similar item clusters are generated, then to remove the sparsity in the user-item matrix and to predict rating for the active user in the system who has not given rating to every item in the system are proposed.
- (iv) For the users already registered into the system, their representation or their profile is generated based on various features like LOD (Linked Open Data) and social network graph feature.
- (v) Similarity between the users is then calculated by not using traditional similarity measures as they have several drawbacks; rather, a modified technique is proposed for better outcome.
- (vi) When *new user* enters into the system using their Facebook ID and DBpedia ID (if any), automatically system generates his profile using *User Profile Generation Module*.
- (vii) Then classifier is trained with data such as user's profile features and class label as "user's cluster."
- (viii) So, when new user's profile is given to classifier for prediction of user cluster to which new user may belong, classifier will predict the corresponding user's cluster.

- (ix) Once we find the "User cluster" to which new user belongs to, system then analyzes the rating provided to each "Item cluster" by only those users who are present in this predicted "User cluster."
- (x) The average weight of the rating given to each Item cluster by the user present in the predicted cluster is calculated.
- (xi) Item cluster with highest rating value is recommended to the new user.

4.1. Item-Based Clustering. In this module, similarity between two items is calculated based on their domain-specific ontology and based on explicit rating provided by the user.

4.1.1. Item Similarity Calculation Based on Ontology. Ontologies provide vast information in any domain which could be very beneficial in the recommendation system. Most of these researchers have considered only one attribute to calculate item similarity based on ontology, and they somehow have neglected the multilevel and complex structure of ontologies. For example, many researchers have only used "genre" of a movie to find the similar set of movies based on ontology. As an abstract sample of the domain ontology illustrated in Figure 2, we define C as the recommended item class, which is used as the suggested target class. The class contains two attributes, $A1$ and $A2$, and a subclass SC , which itself has its attributes, $A3$, $A4$, and $A5$. For example, in a movie recommendation system, if C was the movie class, then $A1$ and $A2$ could denote the copyright and release date, while SC could be the "Movie_origin" having its own attributes, $A3$, $A4$, and $A5$, representing, for instance, Asia, Europe, and North America, respectively.

Ontologies represent the semantic description of any domain, so calculating the similarity between the two items based on their ontology is a very crucial task. In this research, item-based semantic similarity is calculated using binary Jaccard similarity coefficient. For two items to be similar, their own attributes as well as their subclass's attributes need to be similar. So, to find the similarity, for an item class C which has an attribute " At " and its value could fall in m categories, each item is considered as a binary vector $V_c = (v_{c,1}, v_{c,2}, \dots, v_{c,m})$ where a binary variable $v_{c,p}$ ($p = 1, \dots, m$) is defined as follows:

$$\begin{aligned} V_c, p &= 1, & \text{if item belongs to category } p, \\ V_c, p &= 0, & \text{if item does not belong to category } p. \end{aligned} \quad (1)$$

Then, the semantic similarity of items x and y for attribute " At " is presented as follows [33, 34]:

$$SSim(C_i \cdot At, C_j \cdot At) = \frac{T_{11}}{T_{11} + T_{01} + T_{10}}, \quad (2)$$

where T_{01} , T_{10} , and T_{11} , respectively, indicate the total number of categories for $(v_{ci,At}=0; v_{cj,At}=1)$, $(v_{ci,At}=1; v_{cj,At}=0)$, and $(v_{ci,At}=1; v_{cj,At}=1)$.

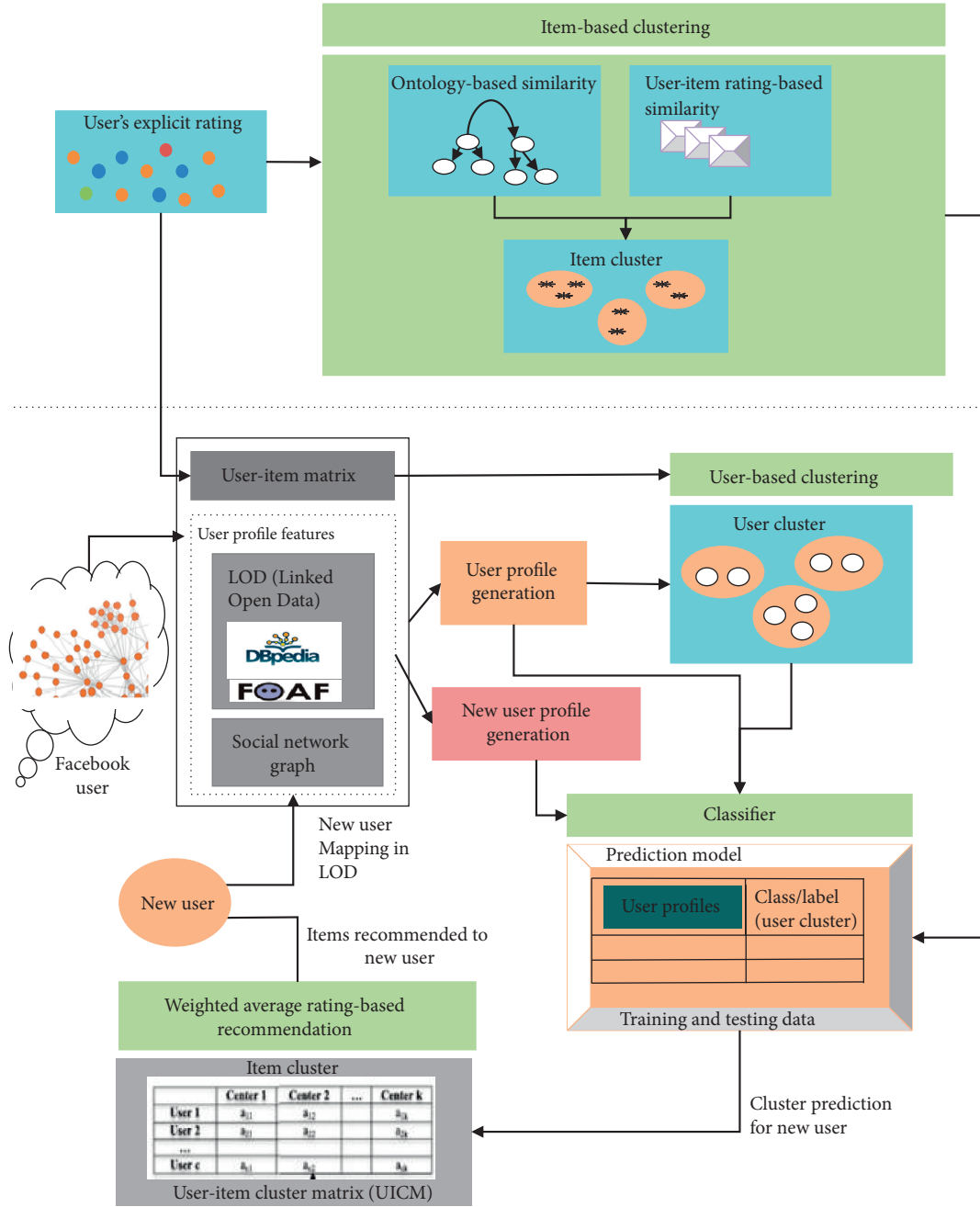


FIGURE 1: Proposed system framework.

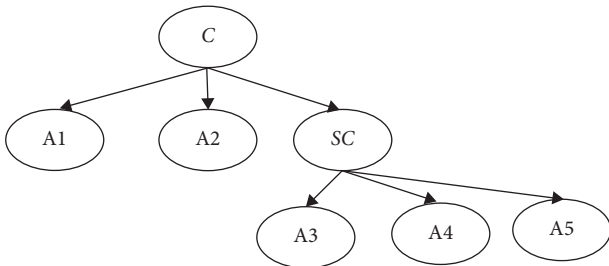


FIGURE 2: Sample ontology for item.

Illustration. Consider the movie having one of its attribute “genres” like comedy, romantic, fiction, drama, and horror as represented in Table 1. Therefore, the categories in which this attribute belongs to are 6, and therefore, the value assigned to “ p ” is 6.

So, for each item, their respective vectors can be represented as follows:

$M1 = (0, 1, 1, 1, 0, 0)$, $M2 = (1, 0, 0, 1, 0, 0)$, $M3 = (0, 0, 1, 0, 1, 0)$, $M4 = (1, 1, 0, 1, 0, 0)$, and $M5 = (0, 0, 1, 0, 1, 0)$, where $M1$, $M2$, $M3$, $M4$, and $M5$ are the binary vectors

TABLE 1: Movie-genre matrix.

Movie	Genre					
	Comedy	Romantic	Fiction	Drama	Horror	Science
M1	0	1	1	1	0	0
M2	1	0	0	1	0	0
M3	0	0	1	0	1	0
M4	1	1	0	1	0	0
M5	0	0	1	0	1	0

for movie 1, movie 2, movie 3, movie 4, and movie 5, respectively.

The similarity between pairs of movies can be calculated using equation (2) as follows:

$SSim(M1, M2) = 1/(1 + 1 + 2) = 1/4 = 0.25$, $SSim(M1, M3) = 1/(1 + 1 + 1) = 1/3 = 0.33$, $SSim(M1, M4) = 2/(2 + 1 + 1) = 2/4 = 0.5$, $SSim(M1, M5) = 1/(1 + 1 + 2) = 1/4 = 0.25$, $SSim(M3, M5) = 2/(2 + 0 + 0) = 2/2 = 1$, and $SSim(M2, M4) = 2/(2 + 1 + 0) = 2/3 = 0.66$, where $SSim(M1, M2)$ is the similarity between movie 1 and movie 2 for an attribute “genre.” Correspondingly, the similarity between the same two items for all other attributes in the ontology can be calculated using equation (2).

Unlike the traditional method, complex and multilevel data structures like ontologies as shown in Figure 2 need to be handled, which contain attribute of a class and also attributes of those attributes which itself is a class (i.e.,

subclass). In the sample set of class “C” shown in Figure 2, it has three attributes, with the third attribute “SC” which itself is a class having three more attributes. The formula for calculating the overall semantic similarity between items I_i and I_j having the same ontology containing classes, subclasses, and attributes is explained below in detail.

In this method, all the attributes corresponding to subclasses SC_i and SC_j of items I_i and I_j , respectively, will be analyzed to calculate the semantic similarity between them and then the attributes of class C_i and C_j of items I_i and I_j , respectively. Using recursive computation, the average of the values is calculated, to obtain the similarity between items I_i and I_j until reaching the maximum depth which could be set in the beginning. So, *similarity based on ontology* is calculated using the following equation:

$$S_{ontology}(I_i, I_j) = \frac{\sum_{k=1}^n SSim(C_i \cdot At(k), C_j \cdot At(k))}{n}, \quad (3)$$

where $S_{ontology}(I_i, I_j)$ is the *similarity based on ontology* between items I_i and I_j , if there is no attribute in the ontology which itself is a subclass, $SSim(C_i \cdot At(k), C_j \cdot At(k))$ is the semantic similarity between classes C_i and C_j of two items I_i and I_j for a particular attribute “At,” respectively, and n is the total number of attributes in the ontology of item domain.

If there are attributes in the ontology, which is also a subclass having its own attributes, $S_{ontology}(I_i, I_j)$ can be calculated using the following equation:

$$S_{ontology}(I_i, I_j) = \frac{\sum_{k=1}^n SSim(C_i \cdot At(k), C_j \cdot At(k)) + \sum_{p=1}^m SSim(SC_i \cdot At(p), SC_j \cdot At(p))}{n}, \quad (4)$$

where $S_{ontology}(I_i, I_j)$ is the *similarity based on ontology* between items I_i and I_j , if there is attribute in the ontology which itself is a subclass, $SSim(SC_i \cdot At(p), SC_j \cdot At(p))$ is the semantic similarity between subclasses SC_i and SC_j of two items I_i and I_j for a particular attribute “At” of subclass, respectively, m is the total number of attributes of subclass in the ontology of item domain $1 \leq p \leq m$, $SSim(C_i \cdot At(k), C_j \cdot At(k))$ is the semantic similarity between classes C_i and C_j of two items I_i and I_j for a particular attribute “At,” respectively, and n is the total number of attributes in the ontology of item domain, $1 \leq k \leq n$.

Algorithm 1 explains how to compute similarity based on the ontology of two items represented by common ontology of that particular domain. It takes two inputs as follows:

- (i) Item ontology containing classes, attributes, and relations
- (ii) Set of all items, represented by I .

The output of the algorithm is the *Semantic Similarity Matrix* (SSM) showing the measures of the semantic similarity between two items I_i and I_j based on ontology.

Examples. Suppose Movie_Origin is class of movie ontology, as shown in Figure 3 and different continents are its attributes, and continents can be further considered as subclass having its own attribute.

If both the movies belong to the same “Continent,” then it is required to match this subclass’s attributes like “country,” i.e., country which is the origin of that movie. If both movies belong to India, then it is required to further match their attributes, i.e., which cinema it belongs to like Bollywood, Tollywood, and Punjabi, else it is not required to match their attributes.

4.1.2. Item Similarity Calculation Based on Explicit User Rating. In this module, similarity between items is calculated based on the explicit rating provided by the users in *User-Item Rating Matrix* (UIM) where U is the set of users and I is the set of items, and $r_{u,I}$ is the rating provided by the user u to item I as shown in Table 2. The value of $r_{u,I}$ can vary between 0 and 1.

The ratings given to individual item form the cases, and the similarity between two items is determined by how

Input: Item Ontology $O(C, At, R)$, Set of Items I
Output: Semantic Similarity Matrix, SSM (I, I)
 for each $I_i \in I$
 for each $I_j \in I$
 if ($I_i \text{SC}, I_j \text{SC}$)
 $S_{ontology}(I_i, I_j) = \sum_{k=1}^n SSIM(C_i \cdot At(k), C_j \cdot At(k))/n$
 else
 $S_{ontology}(I_i, I_j) = \sum_{k=1}^n SSIM(C_i \cdot At(k), C_j \cdot At(k)) + (\sum_{p=1}^m SSIM(SC_i \cdot At(p), SC_j \cdot At(p)))/m/n$
 end if
 end for
 end for

ALGORITHM 1: Similarity computation based on ontology.

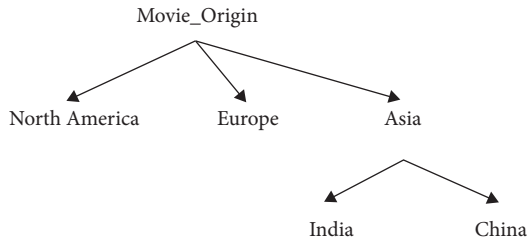


FIGURE 3: Snapshot of movie ontology.

TABLE 2: User-Item Rating Matrix (UIM).

Items	$I1$	$I2$	$I3$	$I4$	$I5$
$U1$	r_{11}	r_{12}	—	r_{14}	r_{15}
$U2$	r_{21}	—	r_{23}	r_{24}	r_{25}
$U3$	r_{31}	r_{32}	—	r_{34}	r_{35}
$U4$	—	r_{42}	r_{43}	r_{44}	r_{45}
$U5$	r_{51}	r_{52}	r_{53}	—	—

similar their rating patterns are provided by the users. The similarity measure used here is as follows:

$$Sim(I_i, I_j) = \sqrt{\sum_{u=1}^n \frac{(I_{iu} - I_{ju})^2}{(I_{iu})^2 + (I_{ju})^2}}, \quad (5)$$

where I_{iu} and I_{ju} are the values of the rating provided to item I_i and item I_j by user u , respectively, and n is the total number of users who rated both the items, $1 \leq u \leq n$. Only the users who rated both the items are used for similarity calculation.

4.1.3. Overall Item Similarity Score. The overall or total similarity score between items would be calculated based on the combination of similarity score provided by ontology using (3) and (4) and explicit user rating using (5) as shown in Figure 4:

$$Similarity(I_i, I_j) = \alpha * S_{ontology}(I_i, I_j) + \beta * Sim(I_i, I_j), \quad (6)$$

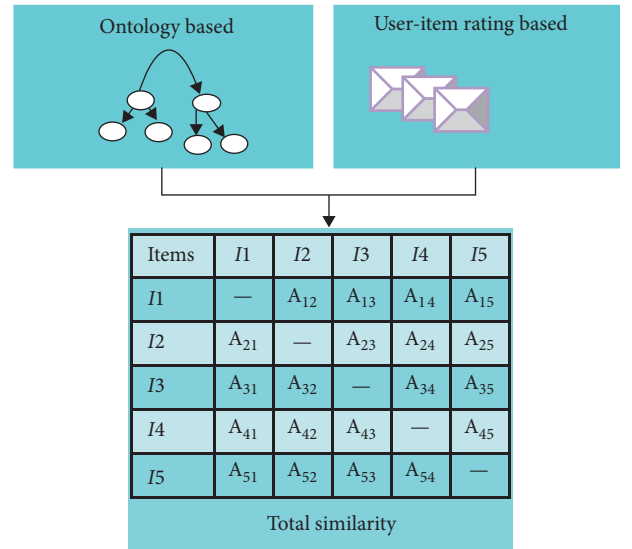


FIGURE 4: Overall item similarity matrix (OSM).

where $\alpha + \beta = 1$, α and β are the control values which could be adjusted by the experts of the domain. In this case, equal weightage is given to both the similarity measures, i.e., $\alpha = 0.5$ and $\beta = 0.5$. $Similarity(I_i, I_j)$ is the overall similarity score between two items, $S_{ontology}(I_i, I_j)$ is the similarity calculated based on ontology, and $Sim(I_i, I_j)$ is the similarity calculated based on explicit user's ratings.

Once the overall similarity is calculated for each item with every other item in the item set using (6), then Overall Item Similarity Matrix (OISM) as shown in Figure 4 is formed which describes the overall similarity score among the items in the item set. In OISM, all rows and columns represent the total number of items in the item set and each cell A_{ij} represents the overall similarity score between item i and item j .

4.1.4. Item Clustering Technique. In this work, fuzzy C means clustering [35, 36] has been used to cluster similar items as it performs well with sparse dataset. In most of the recommendation systems, only few users provide ratings for the items resulting in sparse explicit user-item matrix. In this

research work, both content-based features extracted from ontology and user rating data are considered, since considering only one of them will lead to low accuracy, overgeneralization, and overlapping of the cluster. After creating the clusters, the similar items in the cluster are used for rating prediction task for the target item as described in detail in the next section. Therefore, the items which need to be analyzed are far less than the total numbers of items in the system; hence, it improves the performance of the system [26].

Once the clusters are formed, *User-Item Cluster Matrix (UICM)* where U is the set of user and C is the centers of all item's cluster is generated which represents the value of the average rating provided by user u to item's cluster center j as shown in Figure 5.

The new matrix formed contains the center of each item cluster and the user's rating to each center. In Figure 5, m denotes the number of all users, a_{uj} is the average rating of user u to item's cluster center j , n implies the number of all items, R_{ij} indicates the rating of user u to item j , and k is the number of item centers.

4.1.5. Rating Prediction. For a target item, a sorted list of top T similar items is returned based on the cluster formed (Section 4.1.3). These retrieved values are then used to fill in the empty cells of the target user in *User-Item Rating Matrix*. For each unrated (target) item, the rating is predicted based on the ratings provided by the active user to the items which are similar to that unrated (target) item. The predicted rating is the collective sum of the ratings received by each similar item weighted by the similarity score between that particular similar item and the target item, divided by the sum of similarity scores of the similar items involved. The prediction of the rating of a target user u for an unrated item i is given in the following equation:

$$S_{u,i} = \frac{\sum_{t \in T} \text{Similarity}(i, t) * r_{u,t}}{\sum_{t \in T} \text{Similarity}(i, t)}, \quad (7)$$

where $S_{u,i}$ is the predicted rating, for an item i by user u , $r_{u,t}$ is the rating for similar item t by user u ; $\text{Similarity}(i, t)$ is the similarity score between target item i and item t ; and T is the total number of similar items under consideration.

In some scenarios, it could be possible that there would be no rating given to the top T similar items for a target item by the active user, and in this case, some empty cells would still be present in the user-item matrix after filling in the matrix using (7). So, to overcome this problem, the remaining sparse cells could be predicted using an extended approach. In this approach, for an active user, his rating pattern to rate other items is considered along with the rating provided to the unrated (target) item by other users. The estimated rating of a target user u for an unrated item i using the proposed approach can be calculated as follows:

$$S_{u,i} = \alpha \frac{\sum_{k=1}^K r_{u,k}}{K} + \beta \frac{\sum_{q=1, q \neq u}^n r_{q,i}}{n}, \quad (8)$$

where α and β are the control parameters, where $\alpha + \beta = 1$, $S_{u,i}$ is the predicted rating, K is the number of other items

rated by u (target user), $1 \leq k \leq K$, $r_{u,k}$ is the rating given by u to other items K , n is the number of other users, where $1 \leq q \leq n$, $q \neq u$, who provided rating for target unrated item i , and $r_{q,i}$ is the rating given to target item i by other users q , other than target user u .

α and β are the control values which could be adjusted by the experts of the domain. In this case, equal weightage is given to both the measures, i.e., $\alpha = 0.5$ and $\beta = 0.5$. The algorithm is given below to predict the unrated value in explicit user-item rating matrix, $\text{UIM}(U, I)$, where U is the set of users and I is the set of items. This algorithm will help in removing the sparsity in the $\text{UIM}(U, I)$, and thus, the output of this algorithm is *Dense User-Item Rating Matrix*, $\text{DUIM}(U, I)$ containing no sparsity, as described in Algorithm 2.

Let us take an example to describe in detail the working of Rating Prediction module. In this example, a sample dataset of user-item matrix is considered with unrated values, creating sparsity problem.

Example 1. Consider example of explicit rating provided to each item by each user, which can be represented in a matrix as shown in Table 3.

Suppose this is the matrix containing rating given to movies by different users. Here, “—” represents the sparsity, which means the user who has not rated that particular movie.

So, the similarity calculation based on user's explicit rating for two items $I3$ and $I1$ can be calculated using (5):

$$\text{Sim}(I3, I1) = \text{Sqrt}[\frac{(5-3)^2}{(5^2+3^2)} + \frac{(5-4)^2}{(5^2+4^2)} + \frac{(5-5)^2}{(5^2+5^2)}] = 0.59$$

$$S_{\text{ontology}}(I3, I1) = 0.33, \text{ as calculated in Section 4.1.1}$$

So, the overall similarity calculation based on ontology similarity and explicit user's rating can be calculated using (6):

$$\text{Similarity}(I3, I1) = S_{\text{ontology}}(I3, I1) + \text{Sim}(I3, I1))/2 = 0.46$$

Similarly, we can calculate the overall similarity between two items $I3$ and $I2$, as follows:

$$\text{Sim}(I3, I2) = \text{Sqrt}[\frac{(4-4)^2}{(4^2+4^2)} + \frac{(5-4)^2}{(5^2+4^2)}] = 0.15$$

$$S_{\text{ontology}}(I3, I2) = 0.25$$

$$\text{Similarity}(I3, I2) = S_{\text{ontology}}(I3, I2) + \text{Sim}(I3, I2))/2 = 0.2$$

Similarly, we can calculate the overall similarity between two items $I3$ and $I4$, as follows:

$$\text{Sim}(I3, I4) = \text{Sqrt}[\frac{(3-4)^2}{(3^2+4^2)} + \frac{(5-3)^2}{(5^2+3^2)} + \frac{(4-5)^2}{(4^2+5^2)} + \frac{(5-5)^2}{(5^2+5^2)}] = 0.41$$

$$S_{\text{ontology}}(I1, I3) = 0.5$$

$$\text{Similarity}(I3, I4) = S_{\text{ontology}}(I1, I3) + \text{Sim}(I1, I3))/2 = 0.45$$

Items most similar to item 3 are Item 1 and Item 4, so sparsity calculation to predict the rating value for item 3 by user 1 would be calculated using (7):

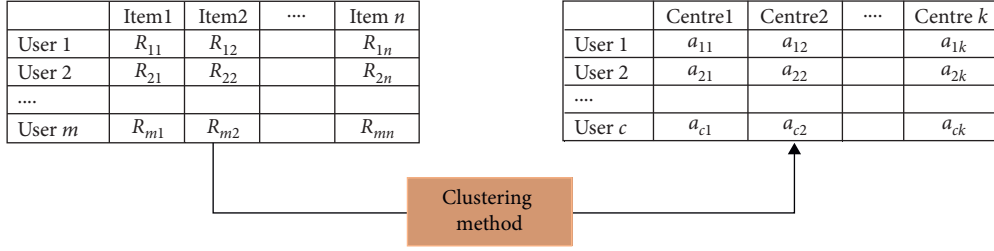


FIGURE 5: User-item matrix (UIM) converted to user-item cluster matrix.

Input: Overall Item Similarity Matrix OISM (I, I), *User-Item Rating Matrix*, UIM (U, I), User Set U , Item Set I
Output: Dense *User-Item Rating Matrix*, DUIM (U, I)
 for each user $u \in U$
 for each (item not rated by u) & epsiv; I
 if (rating given for similar item set)
 $S_{u,i} = \sum_{t \in T} \text{Similarity}(i, t) * r_{u,t} / \sum_{t \in T} \text{Similarity}(i, t)$
 else
 $S_{u,i} = (\sum_{k=1}^K r_{u,k} / K) + (\sum_{q=1, q \neq u}^n r_{q,i} / n)$
 end if
 end for
end for

ALGORITHM 2: Rating prediction.

TABLE 3: Example of *User-Item Rating Matrix* (UIM).

	Item 1	Item 2	Item 3	Item 4
User 1	5	5	—	5
User 2	5	—	3	4
User 3	3	4	—	3
User 4	—	—	5	3
User 5	5	4	4	5
User 6	5	4	5	5

$S_{13} = (0.46 * 5 + 0.45 * 5) / (0.46 + 0.45) = 5$, where S_{13} is the predicted rating for item 3 by user 1 to remove sparsity. Similarly, we could predict the rating value provided to item 3 given by user 3, as follows: $S_{33} = (0.5 * 4 + 0.7 * 3) / (0.5 + 0.7) = 3.41 \geq 3$, where S_{33} is the predicted rating for item 3 by user 3 to remove sparsity. Therefore, $\text{UIM}[3, 3] = 3$; similarly, other values can be predicted.

4.2. User Profile Features. In this section, three families of features to create user's profile are described. Features are extracted by analyzing the social network features, collaborative features, and the features extracted from the information coming from the LOD cloud. *The use of features directly extracted from the LOD cloud is one of the distinguishing aspects of this work.*

4.2.1. Social Network Analysis Features. Social network analysis (SNA) is the field in which complete social configuration is analyzed or processed using traditional network and graph theories. A network structure is composed

of nodes (directional or unidirectional) and edges or links. Node could describe any person or thing; item depending upon requirement and links or ties describes the links between them. A sample social network is shown in Figure 6.

Some useful network parameters are as follows:

- (i) *Centrality.* In network theory, centrality is defined as a factor which recognizes the most significant nodes in the given network. Community structure describes the property of low cohesion and high coupling for the grouping of nodes in the given network. Node is termed as more central if it has higher degree.
- (ii) *Clustering Coefficient* is defined as measurement of the degree by which nodes can cluster together in the given graph. A graph $G = (V, E)$ formally contains set of vertices and edges. Edge represents the connection between two vertices. For example, an edge e_{ij} connects vertex v_i with vertex v_j .

The neighbourhood N_i for a vertex v_i is defined as those neighbours which are directly connected to it.

$$N_i = \{v_j : e_{ij} \in E \vee e_{ji} \in E\}$$

We define K_i as the number of vertices, $|N_i|$, in the neighbourhood, N_i , of a vertex.

- (iii) *Degree for a vertex v :*

- (a) The *in-degree* is “the number of edges started at v . A higher degree node would be more esteemed (choices received).”

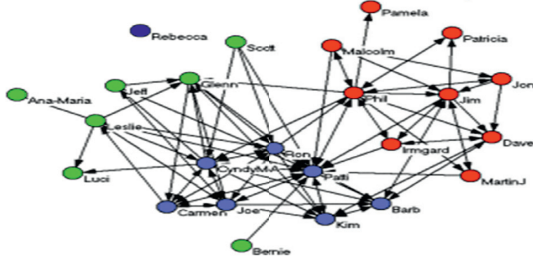


FIGURE 6: An example of social network.

- (b) The *out-degree* is “the number of edges concluded at v . Node degree distribution is one of the most important properties of a graph [10]. The higher degree node would be more central (choices made).”

4.2.2. *Collaborative Features* (c). This class of features models the information encoded in the *User-Item Matrix* thoroughly explained in collaborative filtering (CF) algorithms which contain ratings provided by the users for particular items [37]. Table 4 shows an example representing the collaborative features modeling user’s likes and dislikes. Each user is modeled by extracting the corresponding column vector.

Columns of the matrix represent the inclinations articulated by users, and rows represent rating that each item got from all the users in the system. In many existing research works, various similarity measures such as Pearson correlation, cosine similarity, and Jaccard coefficient had been widely used. But these methods are inefficient in dealing with dataset in which there are more sparsity and very less rating provided by user. Liu et al. [24] presents a new model to find the user similarity with each other. This model focuses on both local and global contexts while analyzing user’s preferences.

There are various drawbacks in calculating similarity using traditional methods; for example, the similarity will be more convincing, if both users have given ratings to more common items. It is very likely that different people have different tendencies of giving ratings as some users always used to give rating on higher side even though they do not like the item so much and vice versa. The traditional similarity measures do not take this thing into account. Many researchers had encoded user’s rating’s absolute value into 1, if user “likes” item, and zero, if user “dislikes” item, but this strategy will eventually become complex while finding similar users.

In this work, method is modified to get better results. The final similarity value is divided with the summation of average rating of each item. The improved method is called as AWPSS (average weighted Proximity-Significance-Singularity).

In equations (9)–(11), the first factor, *Proximity* factor, only considers the difference between two ratings. Next factor *Significance* gives more importance to the ratings if the distance between the two ratings from the median rating is more. Third factor is *Singularity*; it

TABLE 4: Example of a matrix modeling users’ likes and dislikes.

	U1	U2	U3	U4	U5	...	Un
Inception	1	0	1	—	—	...	1
Spiderman	0	1	0	1	1	...	—
Blade runner	—	0	1	—	0	...	0
Step sisters	1	—	1	0	1	...	1

specifies how other ratings are different from these two ratings:

$$\text{Proximity}(r_{u,p}, r_{v,p}) = 1 - \frac{1}{1 + \exp(-|r_{u,p} - r_{v,p}|)}, \quad (9)$$

$$\text{Significance}(r_{u,p}, r_{v,p}) = \frac{1}{1 + \exp(-|r_{u,p} - r_{med}| \cdot |r_{v,p} - r_{med}|)}, \quad (10)$$

$$\text{Singularity}(r_{u,p}, r_{v,p}) = 1 - \frac{1}{1 + \exp(-|r_{u,p} + r_{v,p}/2 - \mu_p|)}, \quad (11)$$

where μ_p is the average rating of item p . $r_{u,p}$ is the rating of item p by user u . Each factor belongs to $(0, 1)$ in our model.

4.2.3. *Linked Open Data- (LOD-) Based Features*. In this work, all the considered features to represent user profile describe numerous facets of the users, since all the information from the social community is a social graph-based; collaborative features are encoded in the description of user’s profile. However, further extending this vector by using LOD attributes is one of the novel approaches of this work to generate user’s profile. The LOD cloud is standard and an important source to acquire descriptive features to model the users.

For example, suppose the recommendation system is for clothing domain, then to find the similarity between users, it depends more on their height, weight, but size rather than their location, occupation, etc. Only DBpedia describes and has standardized these attributes while defining any user’s profile. Although DBpedia contains information to well-known people only, it has been linked with other many databases as well which further provides information about users. In future, as the Linked Data grow, more and more information would be available covering users from all over the world, where each user will be uniquely identified by URI (Uniform Resource Identifier). Most of the domains existed currently are closed domains, as they do not share information among each other. With the growing benefits provided by Linked Open Data, many domains in future might collaborate and share information among each other and contribute in further growth of LOD.

Also, in DBpedia, information about the users or any other items is freely available in the LOD cloud in RDF format. Using SPARQL query endpoint, this information can be easily be extracted by providing only two types of

information: the URI of the resource needed and the name of its attribute to get the desired information.

To obtain information from the LOD, it is highly required to know about the URI of the resource. To extract data about any item such as movie and music, mapping needs to be done, to identify the required item to the corresponding item in linked database. It would be the only entry point to the LOD. The technique for mapping items in LOD is out of scope of this work.

DBpedia contains information about millions of user at present, which will grow in the future as well, thus allowing users from other isolated and famous social networks like Facebook and Twitter to be a part of Linked Open Data initiative. User's privacy and authentication issues would play a major role in linking the data from around the world. Likewise, FOAF (Friend-of-a-friend) files have become widespread and now act as a backbone containing social metadata of various sites such as PeopleLink and LiveJournal.

In our approach, it is very crucial to consider the number of facets to represent user depending upon domain rather than just analyzing user's demographic information such as age, gender, and occupation only. In order to gather LOD-based features related to each user, an explicit requirement from the user side would be needed while registering in the system. They have to specify their DBpedia ID (if any) and also they need to login in the application using their Facebook ID ("Log In as Facebook User") and need to take permission from user to access basic information from his profile. If the desired information is not available in DBpedia, some basic information about the user can be gathered from his social profile as shown in Figure 7.

Next, for each domain, a subset of relevant properties is defined, and finally, we have used SPARQL or Facebook 4j API to extract the required data. Figure 8 shows how to create "first app" and create user authentication token using Graph API Explorer. Also, Figure 9 shows the attributes whose data can be extracted using this API, provided user has granted permissions to do so.

To represent the feature corresponding to each user, a *vocabulary* of LOD-based features is built. The value of each feature in the vocabulary was set to 1 if the user is described through that RDF property and 0 otherwise.

These detailed features used to represent user are very beneficial for the proposed *data model*, as different sources provide different aspects related to a user such as how is user's rating pattern, the importance of user among the community as represented by in-degree and centrality, and vast number of features provided in LOD cloud. Entire features club together to provide comprehensive and effective representation of user's profile.

Table 5 shows some of the attributes representing various facets of user and they are gathered from DBpedia LOD cloud. Each feature is represented using two things: [*property*, *value*], and different users could have the same or different value for each entity.

4.2.4. User Clustering. In this module, similarity between the users is calculated based on the features, i.e., by

analyzing the social network features, collaborative features from *Dense User-Item Matrix (DUIM)*, and the features extracted from the information coming from the LOD cloud.

The number of features encoded in each group can vary depending on the items in the recommender system and the domain of the recommender system. It can be more clearly described in Table 6.

Each user is represented by binary vector, containing 0s and 1s, depending on whether that user falls into that feature or not. Each user in user set is considered as a binary vector $U = (u_{f1}, u_{f2}, \dots, u_{fm})$ where a binary variable u_f ($f = 1, \dots, m$) is defined as follows:

$$\begin{aligned} u_f &= 1, & \text{if user belongs to that } p, \\ u_f &= 0, & \text{if user does not belong to that } p, \end{aligned} \quad (12)$$

where m is the total number of features contained in collaborative and LOD-based feature group.

The similarity between each user to other corresponding users in the user set is presented in the following equation:

$$U_{sim}(U_i, U_j) = \frac{T_{11}}{T_{11} + T_{01} + T_{10}}, \quad (13)$$

where T_{01} , T_{10} , and T_{11} , respectively, indicate the total number of categories for $(u_{if} = 0; u_{jf} = 1)$, $(u_{if} = 1; u_{jf} = 0)$, and $(u_{if} = 1; u_{jf} = 1)$.

Example. In this example sample, vector for user 1, user 2, and user 3 is represented for collaborative and LOD-based features group as shown in Table 7.

Since the values for *Social Network Graph* feature cannot be binary values, i.e., 0 and 1, so to find the similarity of users based on social network graph features, simply normalization of the vectors using $L1$ norm or $L2$ norm can be performed and then Euclidean distance can be applied to find the similarity between users based on social graph features.

For example, if the two users have more in-degree and centrality, it means they are more similar to each other. The overall similarity between the users can be calculated by taking the average of the similarity scores obtained from each feature group.

In this work, fuzzy C means clustering [35] has been used to cluster similar items as well as users. FCM provides soft clustering in which each data point provided with the membership value which describes how much that point belongs to that particular cluster. Also, it works best for large dataset with many features or dimensions. In this work, three groups of features are considered for generating user clusters, since considering only one of them will lead to low accuracy, overgeneralization, and overlapping of the cluster. After creating the user clusters, the similar users in the cluster are used for recommending items to *new user*. Thus, it results in improving performance since the cluster that should be analyzed includes much fewer users compared to the number of all items [38].

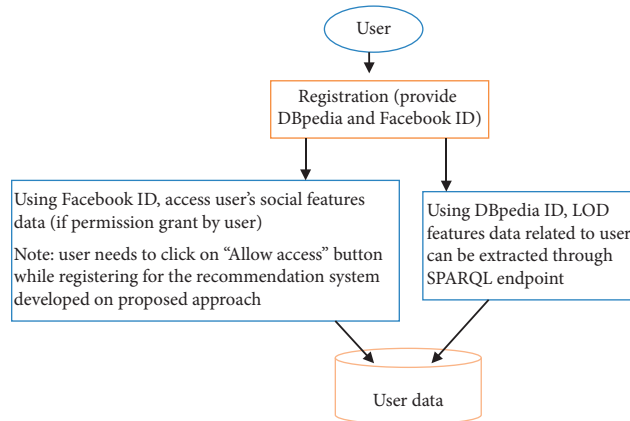


FIGURE 7: Workflow diagram of populating information about user.

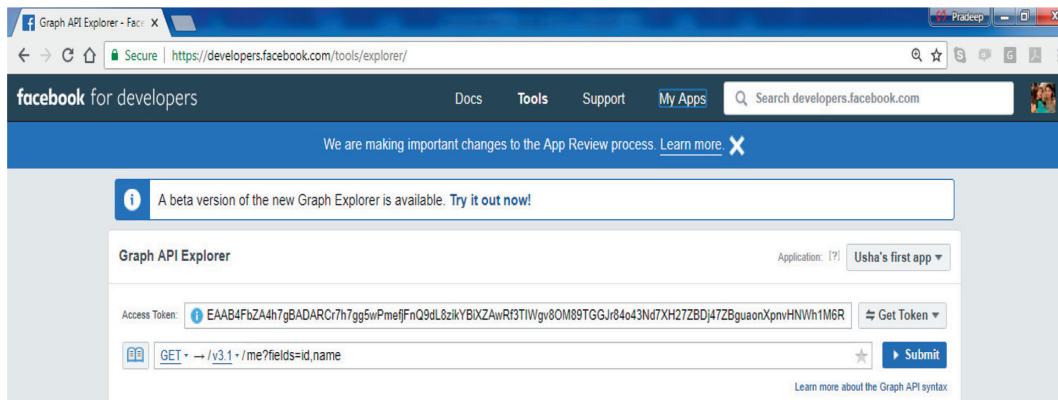


FIGURE 8: Graph API Explorer.

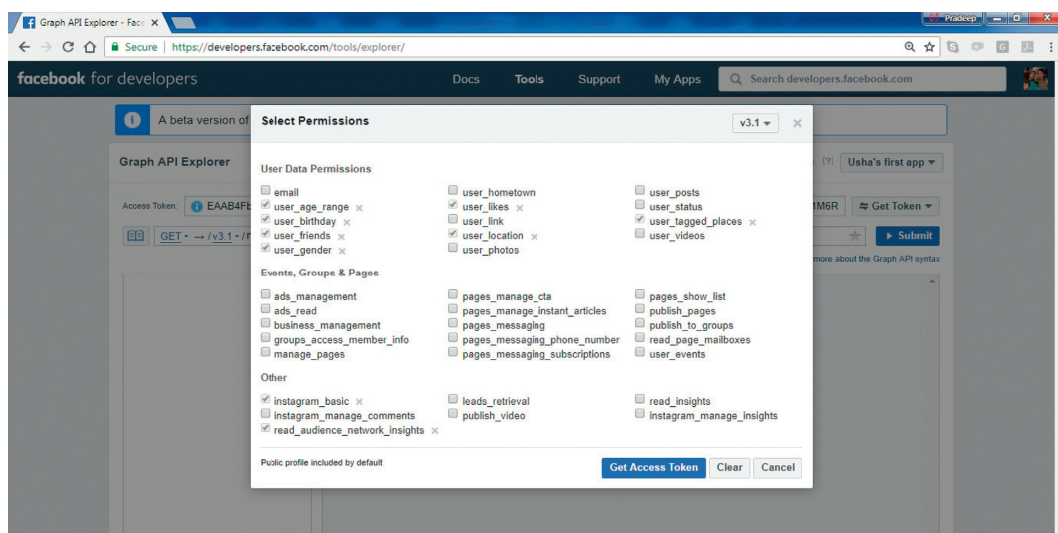


FIGURE 9: Facebook's user account permissions.

TABLE 5: Partial representation of the vector modeling the LOD-based features extracted from DBpedia for the users User 1, User 2, and User 3.

Basic LOD-based features	User 1	User 2	User 3
(dbo:birthPlace, dbr:Germany)	1	0	0
(dbo:birthPlace, dbr:India)	0	1	1
(dbo:birthSign, dbr:Scorpion)	0	0	1
(dbo:College, dbr:IIT Delhi)	0	1	0
(dbo:Employer, dbr:Amazon)	1	0	0
(dbo:age, dbr:36)	1	0	1
(dbo:age, dbr:22)	0	1	1
(dbo:Discipline, dbr:Arts)	1	0	0
(dbo:sex, dbr:Female)	0	0	1
(dbo:Profession, dbr:Racer)	1	0	1
(dbo:Profession, dbr:Dancer)	0	1	0
(dbo:tattoo, dbr:Shiva)	1	0	0
(dbo:skinColor, dbr:brown)	0	1	1

TABLE 6: Comparison among the number of features encoded by each group.

S. no.	Feature category	Number of features
(1)	Social network feature	4
(2)	Collaborative feature	No. of items in item set
(3)	LOD-based features	Values vary with domain

TABLE 7: Vector representing users based on features value.

User	Feature group								
	Collaborative features					LOD-based features			
User 1	0	1	1	0	1...	0	1	0	0...
User 2	1	0	1	0	0...	0	1	1	1...
User 3	1	0	0	0	1...	1	0	1	1...

5. Weighted Average Recommendation Framework

In this proposed research work, the pure new user cold-start problem is formulated, which computes top- N recommendations for a *new user*, who just registered in the system and has not rated any item in the system and not even has searched for any items. When a user enters a system, new user profile is generated using profile generation features. Some good performing classification algorithms, namely, *Logistic Regression*, *Random Forests*, and *Naïve Bayes* (NB) algorithm [39], are implemented to analyze the results from each of them and find out the best classification algorithm for this technique. Prediction model is developed for a classifier using various attributes and class labels. Various user profile features are used as attributes for classifier and user clusters such as UC1, UC2, UC3, UC4, and UC5 as class labels. Classifier is trained with 70% of data and rest 30% of data are used as test data. An algorithm is explained which describes in detail the steps taken by the recommendation system to recommend relevant items to the *new user* as described in Algorithm 3.

6. Experimental Results

In this section, two real-world datasets are used to evaluate the proposed recommender system which deals with pure user cold-start problem. Also the results are compared with the existing state-of-the-art recommendation systems.

6.1. Dataset Description. MovieLens dataset: one of the famous datasets used for the evaluation of the recommender system is the MovieLens dataset which is easily available on <http://www.movieLens.org>. There are 6040 and 3952 numbers of users and movies, respectively, in the dataset. Rating had been given on a 5-star scale. Only those users are selected for evaluations who have given at least 20 ratings to the items in the system. Therefore, this dataset includes 1000,209 anonymous ratings based on the number of movies and users.

Yahoo! Webscope R4 dataset: Yahoo! Research Alliance Webscope program provided this dataset (<http://webscope.sandbox.yahoo.com>), in which rating is given on 5-star scale and the dataset is divided into two training and testing datasets. The training set includes 7642 users, 11,915 movies, and 211,231 ratings. The testing set includes 2309 users, 2380 movies, and 10,136 ratings.

In this study, LOD features are extracted from DBpedia SPARQL endpoint (<http://it.dbpedia.org/sparql>) and Facebook 4j API has been used to gather information about the user. Also web crawler WebSPHINX (<http://cs.cmu.edu/rcm/websphinx/>) has been used to crawl content related to items from IMDb (<http://imdb.com>). For these evaluations, randomly 80% of data are used for the training set and the remaining 20% data are used for the testing set.

6.2. Evaluating the Recommender System. The proposed recommender system was developed using PHP (7.1) language, under a 4 GHz processor, 8 GB RAM, and 64-bit Microsoft Windows 2010. The proposed approach is compared and evaluated with various related research works such as CF recommendation engine using Pearson nearest neighbour algorithm, item-based prediction method with clustering, SVD and ontology, and user- and item-based prediction methods with clustering and SVD but with no contribution of ontology from two perspectives including time throughput (recommendation per second) and accuracy.

6.2.1. Experiment 1 (Scalability Analysis). In the first experiment, the effectiveness of the proposed approach which is based on ontology-based similarity, clustering, and LOD is evaluated. Throughput which is defined as the number of recommendations per second is used for evaluation. MovieLens and Yahoo! Webscope R4 datasets are used to show the effectiveness of the proposed method in improving the scalability of the overall system. Figures 10(a) and 10(b) show the performance of the proposed method with state-of-the-art techniques; throughput is presented on y -axis and is plotted as a function of the cluster size. It could be clearly

Input: User Profile, Item Clusters I , User Cluster U , User-Item Matrix

Output: N Items recommended to new user

- (1) When a new user logs in to the system, his profile will be generated by the system using “User Profile Generation” module.
- (2) Then classifier will analyze the user and can predict the “User Cluster” to which this new user belongs to.
- (3) Once the “User Cluster” is found to which new user belongs to, system will analyze the rating provided to each “Item cluster” by only those users who are present in this predicted “User Cluster.”
- (4) The average weight of the rating given to each Item cluster by the user present in the predicted cluster is calculated.
- (5) Item cluster with highest rating value is recommended to the new user.

ALGORITHM 3: Weighted average recommendation.

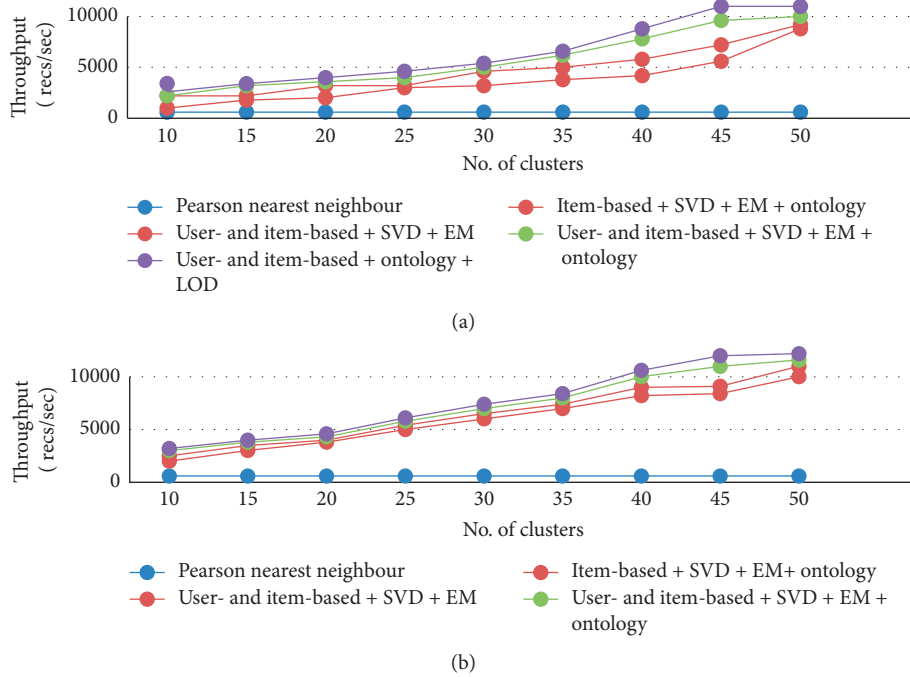


FIGURE 10: (a) Scalability using MovieLens dataset. (b) Scalability using Yahoo Webscope dataset.

inferred from the graph that the throughput of approach based on ontology similarity, clustering, and LOD is slightly higher than that of other approaches. In clustering, only portion of items/users is analyzed by the recommendation system, as opposed to systems using nearest neighbour approach. Therefore, increase in the size of the cluster is not impacting their throughput as it needs to scan all nearest neighbours.

6.2.2. Experiment 2 (Predictive Accuracy). MAE is the statistical metrics to analyze the predictive accuracy. In this experiment, the MAE between the predicted and the actual ratings is measured. MAE is presented as follows:

$$\text{MAE}(\text{pred}, \text{act}) = \sum_{i=1}^n \frac{\text{pred}_{u,i} - \text{act}_{u,i}}{N}, \quad (14)$$

where N is the number of items on which a user u has expressed an opinion. The proposed method using MAE for predictive accuracy is evaluated and compared with Pearson

nearest neighbour algorithm, item-based prediction method with clustering, SVD and ontology, and user- and item-based prediction methods with clustering and SVD. For this evaluation, different numbers of neighbors (k) are considered ($k = 10, 20, 30, 40, 50, 60, 70, 80, 90$, and 100).

Figures 11(a) and 11(b) show MAE for various approaches plotted against different neighbourhood size on two datasets MovieLens and Yahoo Webscope, respectively. It is found that the proposed approach based on item-user ontology, clustering, and LOD performed very well in prediction accuracy compared to other approaches. Also it can be clearly observed that, with the use of ontology, the accuracy of the system is improved, as MAE of item-based + SVD + EM + ontology is lower than the user- and item-based + SVD + EM.

6.2.3. Experiment 3 (Decision-Support Accuracy). In hybrid-based recommender system, decision support accuracy metric plays a very crucial role in analyzing the overall

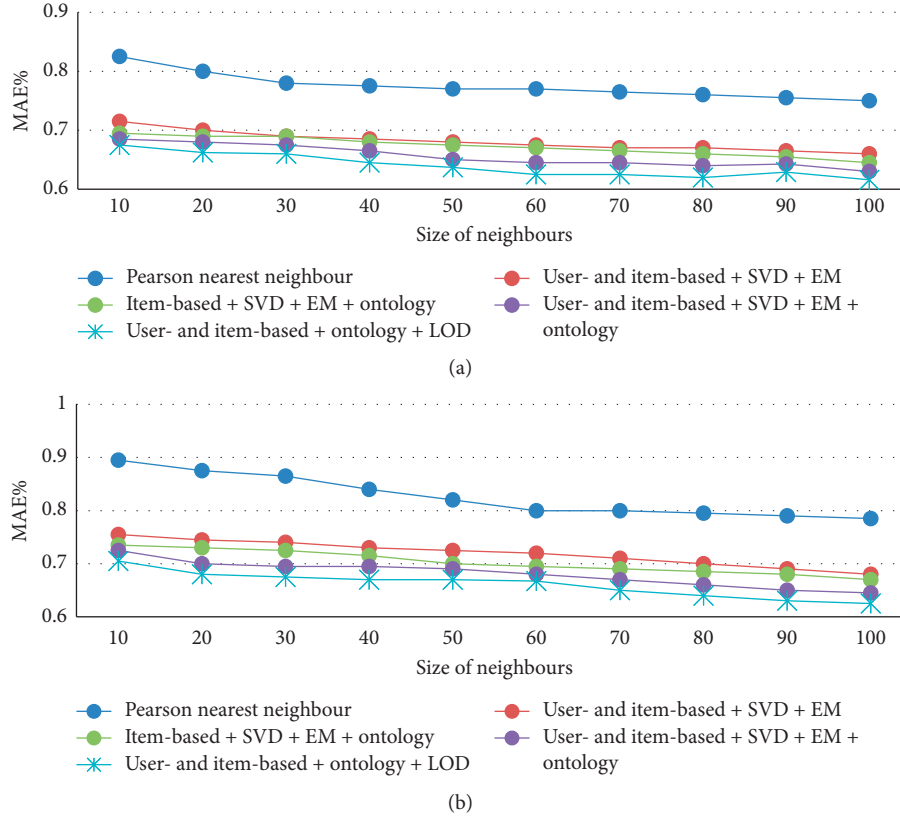


FIGURE 11: (a) MAE using MovieLens dataset. (b) MAE using Yahoo Webscope dataset.

performance of the recommender system. This type of metrics compares the recommended items with the relevant items. The metrics which come under this category are precision, recall, and F -measure.

The precision in equation (15) calculates the fraction of items that are relevant from the list of returned results while the recall in equation (16) calculates the fraction of relevant items that have been retrieved:

$$\text{precision} = \frac{\text{TR}}{\text{TR} + \text{FR}}, \quad (15)$$

$$\text{recall} = \frac{\text{TR}}{\text{TR} + \text{FN}}, \quad (16)$$

where FN is the number of false nonrelevant predictions, TR is the number of true relevant predictions, and FR is the number of false relevant predictions. A metric that considers both values is the F measure (Tsai and Hung 2012) as shown in the following equation:

$$F1 = \frac{(1 + \beta^2) \cdot \text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}, \quad (17)$$

which calculates the mean of the recall and the precision. β can be used to weight the influence of one of both, where $\beta > 1$ raises the significance of the precision, and on the other hand, using $\beta < 1$, influence of recall is increased. So, $\beta = 1$ is considered for a balanced F -measure. The proposed method is evaluated based on different number of top

recommendation, such that $N = 10, 20, 30, 40$, and 50 . Tables 8 and 9 show the $F1$ measures and the precision values for different top- N recommendations. It can be inferred from the table that as compared to nearest neighbour algorithm, precision obtained by the proposed method is significantly high. Also, $F1$ measures are shown in the table, and it can be clearly seen that the $F1$ measures of the proposed system which deal with ontology and LOD have outperformed other methods, majorly nearest neighbour. These results prove that our recommendation system is efficient and scalable as compared to nearest neighbour algorithm. Method A = *User- and Item-based + SVD + EM + Ontology*, Method B = *Item-based + SVD + EM + Ontology*, Method C = *User- and Item-based + SVD + EM*, and Method D = *Nearest Neighbour*.

6.2.4. Experiment 4 (Features Performance). To gain insights about how the set of user profile features will improve the accuracy and performance of the proposed recommendation system, all sets of feature's performance are measured separately and also in some combination. Table 10 shows the result, taking LOD features into account to increase the $F1$ measures as compared to individual features, i.e., demographic-based and social network. Also, it can be seen that combining LOD + SN + D does not provide any major increase in the performance of the recommender system. With the increase in the top- N recommendation system, the performance of the system increased significantly.

TABLE 8: *F1* measures and the precision values for different top-*N* recommendations (MovieLens dataset).

<i>Top N</i>	Proposed system		Method A		Method B		Method C		Method D	
	Precision	<i>F1</i> metric	Precision	<i>F1</i> metric	Precision	<i>F1</i> metric	Precision	<i>F1</i> metric	Precision	<i>F1</i> metric
<i>Top-5</i>	0.803	0.813	0.787	0.797	0.771	0.773	0.719	0.721	0.564	0.583
<i>Top-10</i>	0.806	0.817	0.796	0.807	0.782	0.784	0.736	0.739	0.582	0.601
<i>Top-15</i>	0.822	0.832	0.816	0.827	0.802	0.804	0.747	0.749	0.592	0.615
<i>Top-20</i>	0.833	0.844	0.821	0.833	0.809	0.811	0.757	0.76	0.601	0.622
<i>Top-25</i>	0.846	0.857	0.833	0.844	0.823	0.825	0.769	0.77	0.628	0.65
<i>Top-30</i>	0.839	0.849	0.831	0.84	0.819	0.821	0.757	0.762	0.603	0.605
<i>Top-35</i>	0.832	0.843	0.823	0.832	0.806	0.808	0.75	0.751	0.581	0.59
<i>Top-40</i>	0.829	0.84	0.819	0.83	0.801	0.803	0.739	0.741	0.573	0.579
<i>Top-45</i>	0.821	0.835	0.818	0.827	0.793	0.795	0.733	0.732	0.556	0.558
<i>Top-50</i>	0.819	0.832	0.813	0.822	0.783	0.785	0.723	0.722	0.541	0.546

TABLE 9: *F1* measures and the precision values for different top-*N* recommendations (Yahoo Webscope dataset).

<i>Top N</i>	Proposed system		Method A		Method B		Method C		Method D	
	Precision	<i>F1</i> metric	Precision	<i>F1</i> metric	Precision	<i>F1</i> metric	Precision	<i>F1</i> metric	Precision	<i>F1</i> metric
<i>Top-5</i>	0.768	0.786	0.757	0.767	0.714	0.727	0.669	0.694	0.494	0.516
<i>Top-10</i>	0.786	0.798	0.766	0.777	0.737	0.757	0.683	0.707	0.517	0.534
<i>Top-15</i>	0.803	0.814	0.783	0.792	0.74	0.764	0.688	0.709	0.528	0.549
<i>Top-20</i>	0.808	0.817	0.786	0.797	0.747	0.771	0.692	0.711	0.536	0.557
<i>Top-25</i>	0.802	0.817	0.788	0.797	0.749	0.776	0.697	0.714	0.542	0.565
<i>Top-30</i>	0.824	0.832	0.789	0.801	0.757	0.779	0.704	0.724	0.551	0.571
<i>Top-35</i>	0.826	0.835	0.791	0.803	0.761	0.785	0.715	0.727	0.564	0.582
<i>Top-40</i>	0.825	0.838	0.799	0.805	0.764	0.791	0.721	0.734	0.571	0.594
<i>Top-45</i>	0.827	0.839	0.809	0.814	0.772	0.793	0.727	0.744	0.585	0.608
<i>Top-50</i>	0.839	0.843	0.815	0.821	0.784	0.798	0.739	0.762	0.617	0.631

TABLE 10: *F1* measures and the precision values of feature performance.

<i>Top N</i>	Precision	<i>F1</i> metric	Precision	<i>F1</i> metric	Precision	<i>F1</i> metric	Precision	<i>F1</i> metric
	LOD	LOD	LOD + D	LOD + D	LOD + SN + D	LOD + SN + D	SN	SN
<i>Top-5</i>	0.68	0.76	0.4	0.571	0.4	0.4	0.4	0.4
<i>Top-10</i>	0.68	0.8	0.5	0.5	0.4	0.16	0.4	0.45
<i>Top-15</i>	0.77	0.82	0.33	0.47	0.6	0.48	0.47	0.47
<i>Top-20</i>	0.73	0.84	0.4	0.56	0.55	0.43	0.6	0.6
<i>Top-25</i>	0.85	0.86	0.32	0.6	0.48	0.48	0.6	0.6
<i>Top-30</i>	0.8	0.88	0.67	0.76	0.57	0.33	0.66	0.66
<i>Top-35</i>	0.97	0.97	0.56	0.7	0.63	0.58	0.6	0.75

TABLE 11: Comparison with other research works.

S. no	Publication	Technique used	Advantages	Disadvantages
(1)	An application of fuzzy geographically clustering for solving the cold-start problem in recommender systems [39]	Data like user demographic information, their opinion, and social tags are used to determine the best neighbours for the new user	Analogous users are more accurately determined	Very less user's demographic information is considered
(2)	Semantics-aware recommender systems exploiting Linked Open Data and graph-based features [40]	LOD-based features, Random Forests, Naïve Bayes, and Logistic Regression	Accuracy of recommendation framework is improved	Similarity degree between users is not considered, and sparsity is ignored
(3)	A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques [41]	EM clustering for clustering and nonincremental SVD for dimensionality reduction, Ontology-based similarity	Solve two main drawbacks of recommender systems, sparsity and scalability, using dimensionality reduction and ontology techniques	User's demographic or its browsing information has not been considered

TABLE 11: Continued.

S. no	Publication	Technique used	Advantages	Disadvantages
(4)	A new user similarity model to improve the accuracy of collaborative filtering [24]	Determining the analogous users using new similarity technique clustering algorithms, decision trees	Enhancement in the similarity degrees between users and no requirement for additional data	Needs consideration in choosing the optimal number of groups and the splitting criteria
(5)	Addressing the new user cold-start problem in recommender systems using Ordered Weighted averaging operator [42]	(i) Optimistic exponential type of ordered weighted averaging (OWA) operator is applied (ii) Fusion of CF and demographic and fusion of CF and CBF classifiers have been used	Improvement in performance of the hybrid recommender system under “new user cold-start” problem	Missing values are not handled
(6)	Proposed approach	(i) Ontology-based clustering for item (ii) LOD and social network graph features to build user’s profile	(i) Determination of similar user is more accurate (ii) Improvement in recommendation system’s performance for “pure new user cold-start” problem (iii) Missing values handled, and sparsity prediction is more accurate	(i) Scalability (ii) Filtering features based on domain

TABLE 12: Comparison with other techniques.

S.no	Publication	Dataset	Similarity measure	Environment application	Social network	Ontology used	Linked Open Data	Technique used
(1)	Addressing the user cold start with cross-domain collaborative filtering: exploiting item metadata in matrix factorization [5]	Facebook	Other	Book, movies, music (cross domain)	Facebook	No	DBpedia	Cross-domain collaborative filtering using matrix factorization models
(2)	Latent factor representations for cold-start video recommendation [6]	Video Emotion, Amazon Product	No	Videos (domain independent)	No	No	No	Learning latent factor representation for videos based on modeling the emotional connection between user and item
(3)	Using linked data to build open, collaborative recommender systems [12] An effective recommender algorithm for cold-start problem in academic social networks [7]	Smart Radio	Binary cosine similarity	Music	No	No	DBpedia, Myspace	Used linked data about items for collaborative filtering algorithm
(4)	A method to solve cold-start problem in recommendation system based on social network sub-community and ontology decision model [8]	Created MyExpert app	Others	Academic item	Academic social network	No	No	Enhanced content-based algorithm using social networking
(5)	Exploring social network information for solving cold start in product recommendation [9]	MovieLens	Pearson similarity	Videos	No information	Taxonomy	No	Combining social subcommunity division and ontology decision model
(6)		Douban Website	Other	Books	Douban Website	No	No	Used social network textual information to model user interest and item

TABLE 12: Continued.

S.no	Publication	Dataset	Similarity measure	Environment application	Social network	Ontology used	Linked Open Data	Technique used
(7)	Using semantic web to reduce the cold-start problems in recommendation systems [47]	MovieLens	Composed Similarity, Property Similarity, Jaccard, Jaro Winkler	Movie	No	Yes	No	Used semantic web structure and ontology
(8)	Proposed approach	MovieLens, Yahoo Webscope	New similarity measure and Euclidean distance	Domain independent	Facebook	Yes	DBpedia	Recommendation based on user and item clustering, improved ontology similarity, and taking into account the social network and Linked Open Data features
(9)	Social network data to alleviate cold start in recommender system: A systematic review [32]	Review paper: Systematic combining papers published between 2011 and 2017 on dealing cold-start problems using social network data						

TABLE 13: Evaluation parameters comparison.

S. no.	Publication	Evaluation parameters						
		Precision	Recall	F1 measure	MAE	MMR	Coverage	Others
(1)	Addressing the user cold start with cross-domain collaborative filtering: exploiting item metadata in matrix factorization [5]	Yes	No	No	No	Yes	Yes	Yes
(2)	Latent factor representations for cold-start video recommendation [6]	No	No	No	No	Yes	No	Yes
(3)	Using linked data to build open, collaborative recommender systems [12]	Yes	Yes	No	No	No	No	No
(4)	An effective recommender algorithm for cold-start problem in academic social networks [7]	Yes	Yes	Yes	No	No	No	No
(5)	A method to solve cold-start problem in recommendation system based on social network sub-community and ontology decision model [8]	No	No	No	Yes	No	No	No
(6)	Exploring social network information for solving cold start in product recommendation [9]	No	No	No	No	Yes	No	Yes
(7)	Using semantic web to reduce the cold-start problems in recommendation systems [47]	No	No	No	Yes	No	Yes	No
(8)	Proposed approach	Yes	Yes	Yes	Yes	No	No	Yes

7. Comparison

The comparison in Table 11 compares the proposed approach with various research works done in this area.

The detailed comparison with other techniques is also presented (Tables 12 and 13).

8. Conclusion and Future Scope

The work is contributed to overcome various issues of recommendation system such as accuracy, throughput, sparsity, and new user cold-start problem. To enhance the efficiency of recommender systems, new methods are devised to calculate semantic similarity between two items in

item set based on ontology and for calculating the item similarity based on explicit user rating. In this work, users' similarity is determined using various features like LOD, social network, and collaborative, which can more accurately find similar users and cluster them. The experimental results of the proposed system on two real-world datasets using MAE, precision, and F1 measures show the system to be favorable in improving the throughput, accuracy, decreasing the sparsity, and dealing with new user cold-start problem. In future, work can be investigated in automatic selection of user's feature based on different domains and also various dimensional reduction techniques could be included. Also the impact of all these features can be evaluated on other evaluation metrics such as novelty and diversity [1, 43–46].

Data Availability

The data used to support the findings of this study are included within the article. Readers can access the data supporting the conclusions of the study from MovieLens (<http://www.movieLens.org>) and Yahoo! Webscope R4 datasets (<http://webscope.sandbox.yahoo.com>).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

- [1] H. J. Ahn, "A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem," *Information Sciences*, vol. 178, no. 1, pp. 37–51, 2008.
- [2] M. Y. H. Al-Shamri and K. K. Bharadwaj, "Fuzzy-genetic approach to recommender systems based on a novel hybrid user model," *Expert Systems with Applications*, vol. 35, no. 3, pp. 1386–1399, 2008.
- [3] R. Burke, "Hybrid recommender systems: survey and experiments," *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002.
- [4] C. W.-K. Leung, S. C.-F. Chan, and F.-I. Chung, "An empirical study of a cross-level association rule mining approach to cold-start recommendations," *Knowledge-Based Systems*, vol. 21, no. 7, pp. 515–529, 2008.
- [5] I. Fernández-Tobías, I. Cantador, P. Tomeo, V. W. Anelli, and T. Di Noia, "Addressing the user cold start with cross-domain collaborative filtering: exploiting item metadata in matrix factorization," *User Modeling and User-Adapted Interaction*, vol. 29, no. 2, pp. 443–486.
- [6] S. Roy and S. C. Guntuku, "Latent factor representations for cold-start video recommendation," in *Proceedings of the 10th ACM Conference on Recommender Systems*, pp. 99–106, Boston, MA, USA, September 2016.
- [7] Z. M. Vala Ali Rohani, "An effective recommender algorithm for cold-start problem in academic social networks," *Mathematical Problems in Engineering*, vol. 2014, Article ID 123726, 11 pages, 2014.
- [8] C. Meng and Y. Cheng, "A method to solve cold-start problem in recommendation system based on social network sub-community and ontology decision model," in *Proceedings of the 3rd International Conference on Multimedia Technology (ICMT-13), Advances in Intelligent Systems Research*, Guangzhou, China, November 2013.
- [9] C. Li, F. Wang, Y. Yang, Z. Li, and X. Zhang, "Exploring social network information for solving cold start in product recommendation," in *Web Information Systems Engineering-WISE*, J. Wang, Ed., pp. 276–283, Springer, Berlin, Germany, 2015.
- [10] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, *DBpedia: A Nucleus for a Web of Open Data*, Springer, Berlin, Germany, 2007.
- [11] P. Tomeo, L. Rokach, F. Ricci, and B. Shapira, *Recommender Systems Handbook*, Springer, Berlin, Germany, 2011.
- [12] B. Heitmann, "Using linked data to build open, collaborative recommender systems," in *Proceedings of the AAAI Spring Symposium: Linked Data Meets Artificial Intelligence*, Stanford, CA, USA, March 2010.
- [13] N. Guarino, D. Oberle, and S. Staab, "What is an Ontology?" in *Handbook on Ontologies*, S. Staab and R. Studer, Eds., pp. 1–17, Springer, Berlin, Germany, 2009.
- [14] T. Gruber, *Ontolingua: A Mechanism to Support Portable Ontologies*, Vol. 27, Stanford University; Knowledge Systems Laboratory, Stanford, CA, USA, 1992.
- [15] M. I. Martín-Vicente, A. Gil-Solla, M. Ramos-Cabrer, J. J. Pazos-Arias, Y. Blanco-Fernández, and M. López-Nores, "A semantic approach to improve neighborhood formation in collaborative recommender systems," *Expert Systems with Applications*, vol. 41, no. 17, pp. 7776–7788, 2014.
- [16] C. Porcel, C. Martínez-Cruz, J. Bernabé-Moreno, Á. Tejeda-Lorente, and E. Herrera-Viedma, "Integrating ontologies and fuzzy logic to represent user-trustworthiness in recommender systems," *Procedia Computer Science*, vol. 55, pp. 603–612, 2015.
- [17] S. D. Middleton and N. Shadbolt, "Ontology based recommender systems," in *Handbook on Ontologies*, pp. 447–498, Springer, Berlin, Germany, 2004.
- [18] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific American*, vol. 284, no. 5, pp. 28–37, 2001.
- [19] C. Bizer, "The emerging web of linked data," *IEEE Intelligent Systems*, vol. 24, no. 5, pp. 87–92, 2009.
- [20] M. d. Tomeo, P. Lops, C. Musto, F. Narducci, and G. Semeraro, "Semantics-aware content-based recommender systems," *Recommender Systems Handbook*, pp. 119–159, Springer, Berlin, Germany, 2015.
- [21] J. Wei and A. Fu, "A survey of e-commerce recommender systems," in *Proceedings of the International Conference on Service Systems and Service Management*, IEEE, Chengdu, China, pp. 1–5, June 2007.
- [22] D. Almazro, G. Shahatah, L. Albdulkarim, M. Kharees, R. Msrtinez, and W. Nzoukou, "A survey paper on recommender systems," 2010, <https://arxiv.org/abs/1006.5278>.
- [23] D. Poirier, F. Fessant, and I. Tellier, "Reducing the cold-start problem content recommendation through opinion classification," in *Proceedings of IEEE/WIC/ACM, International Conference on Web Intelligence and Intelligent Agent Technology*, pp. 204–207, Toronto, Canada, 2010.
- [24] H. Liu, Z. Hu, M. Ahmad, H. Tian, and X. Zhu, "A new user similarity model to improve the accuracy of collaborative filtering," in *Knowledge Based Systems*, vol. 56, pp. 156–166, Springer, Berlin, Germany, 2014.
- [25] G. Adomavicius and A. Tuzhilin, "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*, vol. 17, no. 6, pp. 734–749, 2005.
- [26] N. Bassiliades, M. Symeonidis, G. Meditskos, E. Kontopoulos, P. Gouvas, and I. Vlahavas, "A semantic recommendation algorithm for the PaaSPort platform-as-a-service marketplace," *Expert Systems with Applications*, vol. 67, pp. 203–227, 2017.
- [27] A. H. Celdrán, M. G. Pérez, F. J. García Clemente, and G. M. Pérez, "Design of a recommender system based on users' behavior and collaborative location and tracking," *Journal of Computational Science*, vol. 12, pp. 83–94, 2016.
- [28] A. Passant, "DBREC—music recommendations using DBpedia," *ISWC 2010: The Semantic Web-ISWC 2010*, pp. 209–224, Springer, Berlin, Germany, 2010.
- [29] G. Piao and J. Breslin, "Measuring semantic distance for linked open data-enabled recommender systems," in *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, ACM, Pisa, Italy, pp. 315–320, 2016.

- [30] C. Musto, G. Semeraro, P. Lops, M. d. Gemmis, and F. Narducci, "Leveraging social media sources to generate personalized music playlist," in *Lecture Notes in Business Information Processing*, pp. 112–123, Springer, Berlin, Germany, 2012.
- [31] S. Bostandjiev, J. O'Donovan, and T. Hollerer, "TasteWeights: a visual interactive hybrid recommender system," in *Proceedings of the Sixth ACM Conference on Recommender Systems*, ACM, Dublin Ireland, pp. 35–42, September 2012.
- [32] L. A. G. Camacho and S. N. Alves-Souza, "Social network data to alleviate cold-start in recommender system: a systematic review," *Information Processing and Management: An International Journal*, vol. 54, no. 4, pp. 529–544, 2018.
- [33] N. Kermany and S. Alizadeh, "A hybrid multi-criteria recommender system using ontology and neuro-fuzzy techniques," *Electronic Commerce Research and Applications*, vol. 21, pp. 50–64, 2017.
- [34] Q. Shambour and J. Lu, "A trust-semantic fusion-based recommendation approach for e-business applications," *Decision Support Systems*, vol. 54, no. 1, pp. 768–780, 2012.
- [35] A. Baraldi and P. Blonda, "A survey of fuzzy clustering algorithms for pattern recognition. I," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 29, no. 6, pp. 778–785, 1999.
- [36] R. Katarya and O. Verma, "An effective web page recommender system with fuzzy c-mean clustering," *Multimedia Tools and Applications*, vol. 76, no. 20, pp. 21481–21496, 2016.
- [37] J. Herlocker and J. Konstan, "An algorithmic framework for performing collaborative filtering," in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Berkeley, CA, USA, pp. 230–237, August 1999.
- [38] S. Gong, "A collaborative filtering recommendation algorithm based on user clustering and item clustering," *JSW*, vol. 5, no. 7, pp. 746–752, 2010.
- [39] T. Mitchell, *Machine Learning*, McGraw Hill Series in Computer Science, McGraw-Hill, New York, NY, USA, 1997.
- [40] M. Cataldo, P. Lops, M. d. Gemmis, and G. Semeraro, "Semantics-aware recommender systems exploiting linked open data and graph-based features," *Knowledge-Based Systems*, vol. 136, pp. 1–14, 2017.
- [41] M. Nilashi, O. Ibrahim, and K. Bagherifard, "A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques," *Expert Systems with Applications*, vol. 92, pp. 507–520, 2018.
- [42] J. Baisiri, A. Shakery, B. Moshiri, and M. Hayat, "Addressing the new user cold-start problem in recommender systems using ordered weighted averaging operator," *IJICTR*, vol. 2, no. 4, 2010.
- [43] C. Martinez-Cruz, C. Porcel, J. Bernabé-Moreno, and E. Herrera-Viedma, "A model to represent users trust in recommender systems using ontologies and fuzzy linguistic modeling," *Information Sciences*, vol. 311, pp. 102–118, 2015.
- [44] P. Resnick, N. Iacoyou, M. Suchak, P. Bergstorm, and J. Riedl, "GroupLens: an open architecture for collaborative filtering of netnews," in *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, pp. 175–186, Chapel Hill, NC, USA, 1994.
- [45] L. Son, K. Cuong, N. Minh, and N. Canh, "An application of fuzzy geographically clustering for solving the Cold-Start problem in recommender systems," in *Proceedings of the 2013 International Conference on Soft Computing and Pattern Recognition (SoCPaR)*, IEEE, Hanoi, Vietnam, 2013.
- [46] M. Vozalis and K. Margaritis, "Collaborative filtering enhanced by demographic correlation," in *Proceedings of the AIAI Symposium on Professional Practice in AI of the 18th World Computer Congress*, pp. 293–402, Toulouse, France, August 2004.
- [47] O. N. Belloui, "Using semantic web to reduce the cold-start problems in recommendation systems," in *Proceedings of the Second International Conference on the Applications of Digital Information and Web Technologies*, pp. 525–530, London, UK, 2009.