Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A multi-level fusion based decision support system for academic collaborator recommendation

Tribikram Pradhan [*], Sukomal Pal

Department of Computer Science and Engineering, Indian Institute of Technology (BHU), Varanasi, Uttar Pradesh, India

ARTICLE INFO

Article history:
Received 9 April 2019
Received in revised form 11 March 2020
Accepted 17 March 2020
Available online xxxx

Keywords:
Collaborator recommendation
Meta-path analysis
Random walk with restart (RWR)
Topic modeling
Rank-based fusion
LSTM model

ABSTRACT

In academia, researchers collaborate with their peers to improve the quality of research and thereby enhance academic profiles. However, information overload in big scholarly data poses a challenge in identifying potential researchers for fruitful collaboration. In this article, we introduce a multi-level fusion-based model for collaborator recommendation, DRACoR (Deep learning and Random walk based Academic Collaborator Recommender). DRACoR fuses deep learning and biased random walk model to provide the recommendation for potential collaborators that share similar research interests at the peer level. We run a topic model on abstracts and Doc2Vec on titles on year-wise publications to capture the dynamic research interests of researchers. Author–author cosine similarity is computed from the feature vectors extracted from abstracts and titles and is then used to weigh edges in the author–author graph (AAG). We also aggregate various meta-path features with profile-aware features to bias the random walk behavior. Finally, we employ a random walk with restart(RWR) to recommend top *N* collaborators where the edge weights are used to bias the random walker's behavior. Extensive experiments on DBLP and hep-th datasets demonstrate the effectiveness of our proposed DRACoR model against various state-of-the-art methods in terms of precision, recall, F1-score, MRR, and nDCG.

1. Introduction

Recommender systems are used to recommend users different objects based on their personal likings by using various data analysis techniques [1–4]. Generally, academic recommender systems provide recommendations for collaborators [5–7], papers [8–11], citations [12–15] and academic venues [16–18]. These systems have been useful to academicians as they objectively provide users with personalized information services [1,19,20]. Although there have been quite a few works on different academic recommendations, only a little body of work exists in the collaborator recommendations.

Studies suggest that the more the collaboration, the higher is the popularity in the research community. A large number of collaborations are taking place among researchers, and productive students tend to be more collaborative [21,22].

There are various motivations for a researcher to search for collaborators.

(a) Generally, researchers collaborate to improve their research exposure and profile. It also helps them to achieve more influential research and higher productivity.
(b) Junior researchers who are not acquainted with the new research areas may look for collaborators to exchange ideas and; for acquiring expertise and resources for high-quality research.
(c) A veteran researcher may know her research domain exceptionally well; however, when she ventures into another field or works in an interdisciplinary area, she needs to look for potential collaborators.

Of late, tremendous growth in scholarly big data has made it difficult to search for and find out appropriate information [23–25]. For example, DBLP[1] dataset, a collection of scientific publication records and their relationship within that collection has 4,419,797 publications from more than 2,205,561 researchers in 9,585 computer science conferences[2] and more than 4,152[3] journals [26]. This phenomenon has not only created ample opportunities for researchers but has also given rise to new challenges, especially in the area of academic collaboration [27].

To recommend appropriate collaborators, we need to focus our study from the perspective of a researcher's needs, as discussed below.

[*] Corresponding author.
E-mail addresses: tpradhan.rs.cse16@itbhu.ac.in (T. Pradhan), spal.cse@itbhu.ac.in (S. Pal).

[1] http://dblp.uni-trier.de/db/.
[2] http://dblp.uni-trier.de/db/conf/.
[3] http://dblp.uni-trier.de/db/journals/.

https://doi.org/10.1016/j.knosys.2020.105784
0950-7051/© 2020 Elsevier B.V. All rights reserved.

(i) What are the most academic influence-aware features in a big-scholarly network which can affect the co-authorship explicitly or implicitly of target researcher?
(ii) How to capture the rapidly changing nature of the research domains and research interest of individual researchers?

Mainly, there are two types of collaboration recommendations as described below [28,29].

(a) Recommendation of most potential collaborators (MPCs) who have never worked along with the target researcher (i.e., to build new collaborations).
(b) To recommend the most valuable collaborators (MVCs) among researchers who have collaborated with the target researcher before (to reinforce old collaborations).

Numerous collaborator recommendation approaches have been proposed in the past. Initially, there were keyword-based [6], or topic-based [5] that recommended collaborators based on the similarity of the research area. Most of the above-mentioned models fail to incorporate the essence of researchers' collaboration patterns or social proximity leading to inappropriate collaborator recommendations. To resolve the issue of social proximity, a few research works have paid more attention to incorporate network structure for better recommendation [30–32]. However, in content analysis based approaches, the authors' co-authorship profile is not taken into account. On the other hand, network analysis based techniques do not consider content-based similarity and mainly focus on a single type of relation (co-author, citation, or co-citation), and fail when a researcher changes her research interest.

Recently few papers used a hybrid approach [28,29,33] taken of both content and network structure into account. A couple of works have considered two or three features together, to enhance the link importance among researchers in an academic co-authorship network [34,35]. Although they capture the connection and compatibility of collaboration among researchers, there remain a lot of implicit factors like physical distance of their affiliation, age or pedigree, personality that affect collaboration in real life. We do not explicitly consider these features and/or feed enough data to learn their weights in machine learning-based models. The advantage of deep learning is that it uses unsupervised or semi-supervised feature learning and efficient feature extraction algorithms instead of manually acquiring features, and we presume these implicit factors can be taken into account in deep learning-based methods [36].

Also, for a prolific researcher, research interests change with time, and often they diversify. The present techniques are yet to capture the dynamic research interest of a researcher while identifying the current active researchers in a given area. Moreover, various meta-path features such as citations, co-citations, and academic activity among researchers are not yet adequately exploited in collaborator recommendation. Most of the existing approaches mainly focus on recommending possible collaborators without highlighting the most influential collaborators (MICs).

In this paper, we focus on recommending MICs (MPCs+MVCs). We propose a multi-level fusion based system DRACoR (Deep learning and Random walk based Academic Collaborator Recommender). It fuses Meta-path aggregated Random walk based Collaborator Recommendation (MRCR) that finds out MPCs with Deep learning-Boosted Collaborator Recommendation (DBCR) models that find MVCs so that their combination (MICs) can be recommended.

Key contributions of this work are sixfold as described below.

- Proposed model DRACoR works irrespective of researchers' past publication records and is entirely biased towards the current works. Isolated researchers, researchers with less number of co-authors, or researchers with fewer publication records are also getting an equal chance for inclusion in the final recommendation. Hence it can deal with cold-start[4] for a new researcher. We use author–author graph (AAG) instead of the co-authorship network (CN) to address the issue.
- To capture shift in an author's research interest with time, a time-aware inverse logarithmic weighting scheme is proposed. To identify current active collaborators, we also adopt a topic-aware Author2Vec approach considering LDA on abstracts and Doc2Vec on the titles of papers.
- An improved random walk with restart (RWR) based recommender model MRCR is proposed that aggregates various meta-paths features along with H-index based level similarity and percentage of collaboration and captures MPCs.
- To identify the association and collaboration compatibility among researchers, we adopt a deep learning-based collaborator recommendation model DBCR considering Word2Vec and Long Short Term Memory (LSTM) techniques. The technique provides MVCs.
- We propose a multi-level fusion model DRACoR incorporating both MRCR and DBCR models to provide a list of MICs.
- Comprehensive experiments are conducted using two real-world datasets, i.e., DBLP and hep-th to evaluate the performance of the proposed system DRACoR. Our model outperforms several other state-of-the-art collaborator recommendation models with substantial improvements in precision, recall, $F1$-score, MRR, and nDCG.

This paper is organized as follows. We visit the related literature in Section 2. We provide more elaborate problem description in Section 3. Explanation of different features we adopted in our recommendation framework is provided in Section 4. We discuss about data preprocessing and feature extraction in Section 5, and Section 6. We elaborated the model description of MRCR, DBCR, and fusion model DRACoR in Sections 7–9. Experimental details including data description, evaluation matrices and parameter selection are illustrated in Section 10. In Section 11 we reported experimental results. Study of the proposed approach is reported in Section 12. We finally provide our conclusion on DRACoR in Section 13.

2. Related work

Adomavicius and Tuzhilin [3] authored a comprehensive review of recommender systems and suggested mainly three types of recommender systems based on their working principles. In addition, we also include network-based recommendation [20]. We attempt to provide here necessary background in the collaborator recommender systems according to their taxonomy.

2.1. Content-based method

Lee et al. [31] introduced a content-based recommendation system employing research expertise and their professional social networks. Gallapalli et al. [38] proposed a content-based model for collaborator recommendation using the expertise profiles extracted from researchers' publications and academic homepages. Tang et al. [5] introduced a cross-domain topic learning (CTL) model to rank and recommend potential cross-domain collaborators.

Cohen et al. [6] proposed a keywords-based collaborator recommendation model, incorporating both researcher and a set

[4] Cold start issues mainly indicate the new researchers in academia.

Table 1
Comparison of research works on scholarly network analysis.

Research	Meta-path features	Dynamic interest	Research content	Scholarly-aware features	Hidden-relationship	Network
Lopes [37]	No	No	No	No	No	CN
Xia [35]	No	No	No	No	No	CN
Kong [33]	No	No	Yes(Title)	No	No	CN
Xia [34]	No	No	No	Yes	No	CN
Kong [29]	No	Yes	Yes(Abstract)	No	No	CN
Zhou [28]	Yes	No	Yes(Title)	No	No	CN
DRACoR [Proposed]	Yes	Yes	Yes(Abstract+Title)	Yes	Yes	AAG

CN denotes Co-authorship Network (Definition 4), and AAG denotes author–author graph (Definition 5).

of keywords as an input to the system. Yang et al. [39] proposed a weighted topic model for complementary collaborator recommendation. They employed a greedy heuristic algorithm based on the probabilistic topic model. Liu et al. [40] proposed CAAR, which is designed by jointly representing scholars and research topics based on their mutual dependency and extracting scholars underlying characters for high-quality new collaborator recommendation.

2.2. Network-based method

Co-authorship is one of the most tangible and well-documented forms of scientific collaboration [41]. Newman [42] studied many statistical properties of scientific collaboration networks, including the number of papers written by authors, number of authors per paper, number of collaborators that scientists have, existence, and size of a giant component of connected scientists, and degree of clustering in the networks. Barabasi et al. [43] inferred that the dynamic and the structural mechanisms govern the evolution and topology of coauthor networks.

Liu et al. [44] conducted a coauthor network, for which he defined AuthorRank as an indicator of the impact of an individual author in the network. Their results show clear advantages of Page Rank and Author Rank over the degree, closeness, and betweenness centrality metrics. Lopes et al. [30] employed researchers prior to publications and vector space models to recommend collaborators in an academic, social network.

2.2.1. Random walk algorithm

The random walk model is a popular model in recommendation systems. Mohsen et al. [45] investigated a random walk model combining the trust-based and collaborative filtering approaches for the recommendation. Konstas et al. [32] adopted Random Walk with Restart (RWR) integrating the rich information of both authors and co-authorship relations. Backstrom et al. [46] proposed a supervised random walk based on RWR, to predict and recommend links in social networks. Li et al. [34] proposed a system incorporating both authors and co-authorship to recommend collaborators for a target researcher.

Xia et al. [35] extended their previous model ACRec [34] and presented a system exploiting RWR approach to provide a recommendation. Zhou et al. [28] proposed a random walk with restart based collaborator recommendations in a heterogeneous bibliographic network to recommend collaborators. They used a set of meta-paths rules to simplify a heterogeneous network and used a biased edge weighting to recommend collaborators. Kong et al. [29] used a topic clustering model on researchers' publications in each year and fixed the generated topic distribution by a time function to fit the interest dynamic transformation. Kong et al. [33] used a topic clustering model on the title to identify the academic domains of a researcher and then applied a random walk model to compute the researcher's feature vector.

Zhou et al. [47] proposed a model incorporating academic influence aware and multidimensional network analysis methods (AMIN). They mainly used activity-based collaboration relationships, specialty-aware connection, and topic-aware citation fitness for effective collaboration recommendation. Wang et al. [48] proposed a model SCORE utilizing the weak tie relationships to provide a sustainable collaborator recommendation. They incorporated three perspectives such as collaboration output, collaboration duration, and collaboration index to define collaboration sustainability. Sun et al. [49] proposed Career Age-Aware Scientific Collaborator Recommendation (CAASCR) model consisting of three parts: authorship extraction, topic extraction, and career age-aware random walk for measuring scholar similarity.

2.3. Hybrid method

Chen et al. [50] outlined a framework that makes suggestions of collaborators in view of a combination of the network structure similarity and the researcher's research interests between the source and the target authors. Chaiwanarom et al. [7] suggested a collaborator recommendation in interdisciplinary areas of computer science using degrees of collaborative forces, the temporal evolution of research interest, and similar seniority status. Kong et al. [51] proposed a system TNERec fusing both research interest and network structure. Yang et al. [52] proposed a model based on research expertise, researchers' institutional connectivity, and network proximity through SVM-rank fusion strategy.

We have discussed and analyzed the existing literature and concluded with a few state-of-the-art methods related to our proposed model DRACoR. The comparison results are shown in Table 1.

3. Problem statement and other definitions

In this segment, we exhibited the problem description and discussed various notations and terminology. A heterogeneous network is a special kind of information network, which either contains multiple types of objects or multiple types of links.

Definition 1 (*Heterogeneous Information Network*)**.** (HIN) [53,54]. It is defined as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a node type mapping function $\delta : \mathcal{V} \rightarrow \mathcal{A}$ and a link type mapping function $\mu : \mathcal{E} \rightarrow \mathcal{R}$. Each node $v \in \mathcal{V}$ belongs to one particular node type in the node type set $\mathcal{A}$: $\delta(v) \in \mathcal{A}$, and each link $e \in \mathcal{E}$ belongs to a particular link type in the link type set $\mathcal{R}$: $\mu(e) \in \mathcal{R}$. Here both type of nodes $\mathcal{A}$ and type of edges $\mathcal{R}$ depend on the domain in question. Note that both $|\mathcal{A}| > 1$ and $|\mathcal{R}| > 1$.

Due to the complexity of HIN and also to understand the node types and link types clearly in the network, meta level (schema-level) description is provided. So the concept of network schema is proposed to describe the meta structure of a network [55].

Definition 2 (*HIN Schema [53]*)**.** The HIN schema denoted as $\mathcal{S} = (\mathcal{A}, \mathcal{R})$, is a meta template for an information network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a node type mapping function $\delta : \mathcal{V} \rightarrow \mathcal{A}$ and a link type mapping function $\mu : \mathcal{E} \rightarrow \mathcal{R}$, which is a directed graph defined over node types $\mathcal{A}$ and type of edges $\mathcal{R}$.

Please cite this article as: T. Pradhan and S. Pal, A multi-level fusion based decision support system for academic collaborator recommendation, Knowledge-Based Systems (2020) 105784, https://doi.org/10.1016/j.knosys.2020.105784.

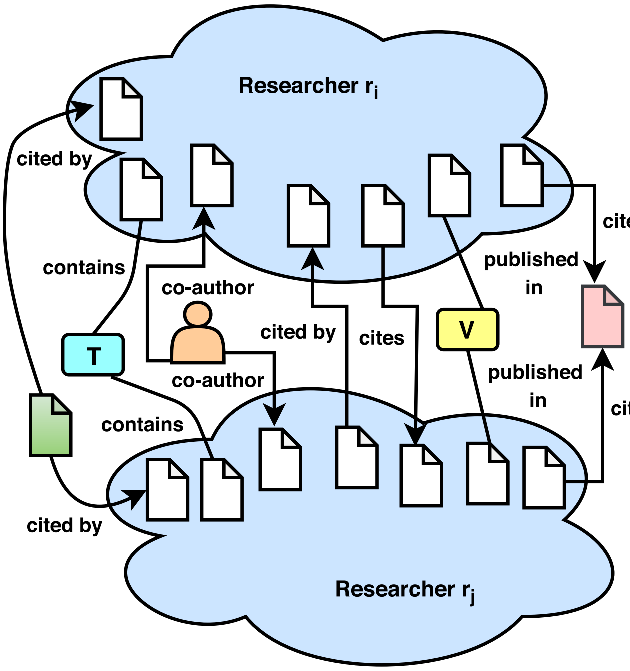


Fig. 1. Graphical representation of SIN graph. *P_main* paper is the paper written by either of the disparate researchers under study and latent metapaths between *P_main* papers may be formed via various vertices types: cited by *P_main* (*P_ref*), cites a *P_main* (*P_cite*), researcher (R), term (T), and venue (V).

Table 2
Type of vertices used in SIN.

No.	Vertices type
1	P_main = {set of papers written by a researcher}
2	P_ref = {set of papers cited by a *P_main* paper}
3	P_cite = {set of papers that cites a *P_main* paper}
4	R (researcher) = {authors of *P_main*, *P_cite*, *P_ref*}
5	T (term) = {terms appearing in *P_main* papers}
6	V (venue) = { Venues of *P_main* papers}

Definition 3 (*Scholarly Information Network (SIN)* [56]**).** SIN graph is an instance of HIN. Here both type of nodes $\mathcal{A}$ and type of edges $\mathcal{R}$ are related to a scholarly network (academia).

Example. In a SIN, $\mathcal{A}$ can be either authors, papers, publication venues, terms etc. Similarly, type of links $\mathcal{R}$ can be any type of relations between a pair of members in $\mathcal{A}$ like paper–paper, author–author, paper–author, paper–venue, author–venue, paper–terms, author–terms, venue–terms etc. In Fig. 1, we show graphical representation of a SIN with all of its vertices types and their relationship. Here we have six type of nodes $\mathcal{A}$, such that $\mathcal{A} = P\_main \cup P\_ref \cup P\_cite \cup R \cup T \cup V$ and seven type of links $\mathcal{R}$ (Table 3). The meaning of each type of node is defined in Table 2. *P_main* paper is the paper written by either of the disparate researchers under study. In Fig. 1, *P_main* papers are those papers written by either researcher r_i or researcher r_j or both. *P_ref* denotes the set of papers cited by a *P_main* paper whereas *P_cite* indicates to a set of papers that cite at least a *P_main* paper.

Definition 4 (*Co-authorship Networks (CN)* [7]**).** Let $G_a = (V_a, E_a)$ be the original co-authorship bibliographic network, with l authors. $V = \{r_1, r_2, \dots, r_l\}$. Each edge $e = (r_i, r_j) \in E$ represents a co-authorship of r_i with r_j in one or more papers.

Table 3
Type of edges used in SIN.

No.	Edges type
1	$n_1 \xrightarrow{written\_by} n_2 : \delta(n_1) \in \{P\_main, P\_ref, P\_cite\}, \delta(n_2) = R, n_1, n_2 \in N$
2	$n_1 \xrightarrow{published\_by} n_2 : \delta(n_1) \in \{P\_main, P\_ref, P\_cite\}, \delta(n_2) = V, n_1, n_2 \in N$
3	$n_1 \xrightarrow{contains} n_2 : \delta(n_1) \in \{P\_main, P\_ref, P\_cite\}, \delta(n_2) = T, n_1, n_2 \in N$
4	$n_1 \xrightarrow{cites} n_2 : \delta(n_1) \in \{P_main\}, \delta(n_2) =$P_ref, $n_1, n_2 \in N$
5	$n_1 \xrightarrow{cited_by} n_2 : \delta(n_1) \in \{P_main\}, \delta(n_2) =$P_cite, $n_1, n_2 \in N$
6	$n_1 \xrightarrow{cites} n_2 : \delta(n_1) \in \{P_main\}, \delta(n_2) =$P_main, $n_1, n_2 \in N$
7	$n_1 \xrightarrow{cited_by} n_2 : \delta(n_1) \in \{P_main\}, \delta(n_2) =$P_main, $n_1, n_2 \in N$

Definition 5 (*Author–author graph (AAG)*)**.** Let $G' = (V', E')$ be the newly generated author–author graph (AAG) from SIN based on the similarity score of abstract and title. $V' = \{r_1, r_2, \dots, r_l\}$. Each edge $e = (r_i, r_j) \in E'$ represents a currently similar research interest of r_i with r_j based on their past publications. There is an edge $e = (r_i, r_j) \in E'$ exists if the similarity score among researcher r_i and r_j is greater than average similarity score. We weight the edges of the network *AAG* using content similarity (linear combination of abstract and title) in order to provide a single score as explained in Section 6.

Example. In Fig. 1, there will be an edge (r_i, r_j) that exists between researcher r_i and researcher r_j if their similarity score will be greater than the average similarity score. In AAG there will be only one type of nodes $\mathcal{A}$ researcher (researcher associated with only *P_main*).

In AAG, two researchers can be connected via different semantic paths, which are called meta-paths.

Definition 6 (*Meta-path* [57]**).** A meta-path $\mathcal{M}$ is a path defined on the SIN graph introduced in Definition 3 . It joins two or more vertices using one or more edges such that $\mathcal{M} = n_1 \overset{l_1}{\rightarrow} n_2 \overset{l_2}{\rightarrow} \dots \overset{l_t}{\rightarrow} n_{t+1}$, where the starting and ending vertices are of same vertex type *P_main*, $\delta(n_1) = \delta(n_{t+1})$ and both belong to *P_main*, $P\_main \in \mathcal{A}$, $\mu(l_1, l_2, \dots, l_t) \in \mathcal{R}$.

Example. In Fig. 1, There will be a meta path between researcher r_i and researcher r_j via the meta path $r_i \overset{writes}{\rightarrow} P\_main \overset{citedby}{\rightarrow} P\_cite \overset{cites}{\rightarrow} P\_main \overset{writtenby}{\rightarrow} r_j$.

Definition 7 (*Random Walk* [58]**).** A random walk is defined as a node sequence $S_r = \{r_1, r_2, r_3, \dots, r_l\}$ wherein the ith node r_i in the walk is randomly selected from the neighbors of its predecessor r_{i-1}.

Definition 8 (*Collaborator Recommendation*)**.** Given a set M of m target researchers $M = \{r_1, r_2, \cdots, r_m\}$,$(m \ll l)$, the collaborator recommendation task is to recommend a list of potential collaborators $K_i = \{r_{i1}, r_{i2}, \dots, r_{in}\}$, $(r_{ij} \in V')$ related to each target researcher r_i where the list is in decreasing order of relevance $(K_i \subset V')$.

A collaborator recommendation problem is essentially a link prediction problem. In an author–author graph (AAG) for a pair of researchers (r_i, r_j), predict whether the node pair can collaborate in the near future (irrespective of the fact whether the pair collaborated earlier or not). However, we are more interested in predicting new collaborators (co-authors) in addition to the existing ones, to the target researcher.

Please cite this article as: T. Pradhan and S. Pal, A multi-level fusion based decision support system for academic collaborator recommendation, Knowledge-Based Systems (2020) 105784, https://doi.org/10.1016/j.knosys.2020.105784.

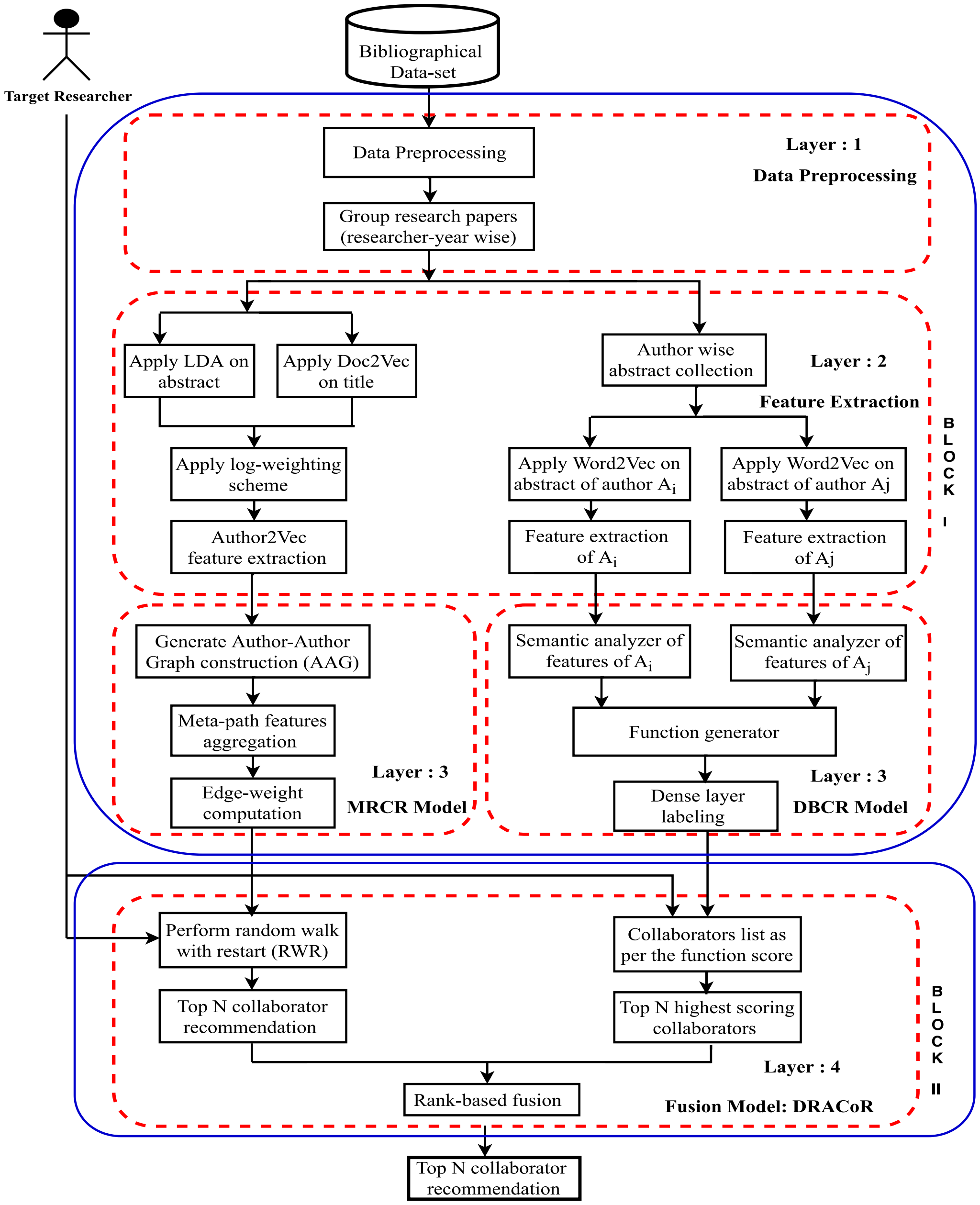

Target Researcher
Bibliographical Data-set
Data Preprocessing
Group research papers (researcher-year wise)
Layer : 1
Data Preprocessing
Apply LDA on abstract
Apply Doc2Vec on title
Author wise abstract collection
Layer : 2
Feature Extraction
Apply log-weighting scheme
Apply Word2Vec on abstract of author A_i
Apply Word2Vec on abstract of author Aj
Author2Vec feature extraction
Feature extraction of A_i
Feature extraction of Aj
BLOCK I
Generate Author-Author Graph construction (AAG)
Meta-path features aggregation
Edge-weight computation
Layer : 3
MRCR Model
Semantic analyzer of features of A_i
Semantic analyzer of features of A_j
Function generator
Dense layer labeling
Layer : 3
DBCR Model
Perform random walk with restart (RWR)
Top N collaborator recommendation
Collaborators list as per the function score
Top N highest scoring collaborators
Layer : 4
Rank-based fusion
Fusion Model: DRACoR
BLOCK II
Top N collaborator recommendation

Fig. 2. Functional architecture of DRACoR.

4. The functional architecture of DRACoR

We propose DRACoR comprised of two blocks: Block-I and Block-II as depicted in Fig. 2. To reduce computational overhead and to make it independent and autonomous target researchers, particularly Block-I is developed once for the entire dataset. Later on, we will utilize the target researcher as an input to interact with Block-II to extract meaningful recommendations from both the MRCR model and DBCR model. We present a layered architecture where each layer realizes a specialized task. The system contains four essential layers, where Layer-1 to Layer-3 have a place with Block-I, and the Layer-4 goes under the classification of Block-II. Four essential layers are portrayed underneath:

(i) *D*ata preprocessing **(Layer-1)**: This step aims to structure the dataset into a formal model for processing. Mainly it is used for faster extraction of researcher-year wise, relevant papers for further use **(BLOCK I)**.

(ii) *F*eature extraction layer **(Layer-2)**: This layer is mainly introduced to extract current research interest by computing Author2Vec approach and extraction of content-aware features of each researcher by Word2Vec approach **(BLOCK I)**.

(iii) *M*RCR model **(Layer-3)**: This model improves the performance of basic random walk based approach by exploiting meta-path features such as previous publication content, citations, co-citations and other similarity (venue, co-author and term) and scholarly influence-aware features such as percentage of collaboration and level similarity in order to recommend highly personalized MVC collaborators **(BLOCK I)**.

(iv) *D*BCR model **(Layer-3)**: To reflect the semantics such as interesting topics shared by two researchers and furthermore to capture hidden relationship among them, LSTM model-based deep learning architecture is incorporated. It utilized previous publications content as word embedding

Please cite this article as: T. Pradhan and S. Pal, A multi-level fusion based decision support system for academic collaborator recommendation, Knowledge-Based Systems (2020) 105784, https://doi.org/10.1016/j.knosys.2020.105784.

layer, and meta-path features are exploited to set dense layer labeling among researchers to recommend MPC collaborators (**BLOCK I**).

(v) Fusion model (**Layer-4**): To provide a diversified personalized recommendation, the MRCR model and DBCR are utilized to firstly make predictions individually and later on a fusion model is applied to integrate the strengths of both the models and to reduce their weaknesses. The outcome acquired from these two models is fused by the standardized Borda count method. We propose this fusion model as DRACoR (**BLOCK II**).

5. Data preprocessing layer (Layer-1)

This step plans to structure the dataset into a formal model for preprocessing. Fundamentally, it is utilized for faster extraction of relevant papers. In the DBLP-citation-network V10[5] dataset, there were 3,079,007 rows and 7 columns. After dropping all the Nan (not a number) values, we left with 2,408,010 rows. We drop all the rows which were left blank in the references column as it will create unnecessary hindrance during training. Similarly, we preprocessed the hep-th (Theoretical High Energy Particle Physics) dataset provided by KDD Cup 2003.[6] After data preprocessing as above, we get 1,922 concurrent authors from 20,961 publications. We have split the authors into different columns with the paper information, and we now have a separate row for authors of every paper. Finally, we removed all the noise in abstracts to get the experimental dataset. The detailed statistics and data collection of DBLP and hep-th datasets are described in Section 10.1.

6. Feature extraction layer (Layer-2)

In this layer, we mainly extract content-based features from the past publications of researchers to compute similarity among them. Generally, the abstract provides a summary containing the main idea of a paper. We use the LDA model on the abstract to generate the feature description [59]. LDA is used for automatically identifying topics and to derive hidden patterns exhibited by a text corpus. We have chosen LDA over other methods to discover coherent topics and their distribution in the abstract. Doc2Vec is used to extract the feature description from the title of a paper as Doc2Vec captures contextual information of words occurring in titles [60]. It is mainly used to generate sentence/document embeddings [61]. It is chosen over other methods due to its potential to overcome the weaknesses such as the ordering of words, semantic of the words, data sparsity, and high dimensionality in bag-of-words models.

Features so extracted from the abstract and the title, are fused in the next layer to use as an input for the MRCR model. The feature extraction for DBCR model is based on a trained Word2Vec skip-gram model with negative sampling, which uses the dataset made using abstracts as the training dataset.

The reason to adopt skip gram model are:

(i) Skip-gram model can capture the semantics for a single word.

(ii) Generally skip-gram with negative sub-sampling outperforms every other methods.

6.1. Topic distribution of research interest

We cluster the abstract of publications for each researcher. To measure a researcher's dynamic research interest, we first build academic documents for each researcher by joining the abstract of the researcher's publication in each year by space. Therefore, for each researcher, we get a set of documents corresponding to each year. Then, we run the LDA model with a special parameter k on the generated documents set which contains the documents from all researchers. The parameter k represents the clustered topic number in LDA. The LDA gives the probability distribution of a researcher's interest over k topics in each year. We treat this as a feature vector of length k.

We follow a similar procedure with publication titles as well. Doc2Vec is used to extract feature vectors from titles in this paper [62]. In this work, vector length for Doc2Vec and LDA have been kept the same to reduce the number of parameters in preliminary experimentation and can be tuned separately in future works.

6.2. Researcher's interest variation with time

Topic distribution of abstract and title embeddings in recent years can describe the current research interest of a researcher more accurately. Hence, to capture the dynamic research interest, we propose a weighted addition of vectors that we get after LDA and Doc2Vec. The vectors in recent years are given more weight, and the weight decays in the decreasing order of the years. For each author, we have two sets of vectors: one from LDA on year-wise abstracts and the other from Doc2Vec on year-wise titles.

The results from LDA and Doc2Vec can be considered as two sets of vectors. L_i^r represents the vector of year-wise topic distribution vectors and D_i^r represents the vector of year-wise title embeddings vectors as depicted in Eqs. (1) and (2). The years considered are 2000, 2001, …, 2012. Each year-wise vector is again a vector of k different topics as given in Eqs. (6) and (7).

$$\boldsymbol{L}_i^r = [L_{2000i}^r, L_{2001i}^r, \ldots, L_{2012i}^r] \tag{1}$$

$$\boldsymbol{D}_i^r = [D_{2000i}^r, D_{2001i}^r, \ldots, D_{2012i}^r] \tag{2}$$

Now, we employ a weighted addition of vectors from each set to get one vector for abstract similarity and one vector for title similarity. We use inverse log-weighting to give more weight to the current year vectors, and the weight reduces in the decreasing order of the years. For each researcher r_i, we get a vector A_i^r for abstract similarity and vector T_i^r for title similarity.

$$\boldsymbol{A}_i^r = \sum_{y_i \in Y} \frac{L_{y_i}^r}{\log_2(y_o - y_i + 2)}, \text{ and} \tag{3}$$

$$\boldsymbol{T}_i^r = \sum_{y_i \in Y} \frac{D_{y_i}^r}{\log_2(y_o - y_i + 2)} \text{ where} \tag{4}$$

$$Y = \{2000, \ldots, 2012\} \text{ and } y_o \text{ is the latest year in } Y. \tag{5}$$

$$L_{y_i}^r = [a_{1i}, a_{2i}, \ldots, a_{ki}] \tag{6}$$

$$D_{y_i}^r = [a_{1i}, a_{2i}, \ldots, a_{ki}] \tag{7}$$

Using A_i^r and T_i^r for a researcher r_i, we compute cosine similarity with their counterpart from the seed paper (r_j) as discussed in next section Author2Vec edge weighting.

[5] https://aminer.org/citation.

[6] https://www.cs.cornell.edu/projects/kddcup/datasets.html.

Table 4
Research topic distribution of researcher r_i.

Year	$Topic_1$	$Topic_2$	$Topic_3$	$Topic_4$	$Topic_5$
2008	0.1	0.2	0.5	0	0.2
2009	0.4	0.2	0.3	0.1	0
2010	0.3	0.1	0.2	0.2	0.2
2011	0.1	0.3	0.1	0.3	0.2
2012	0	0.1	0.3	0.3	0.3

Table 5
Weighted score of topic distribution of researcher r_i.

Year	$Topic_1$	$Topic_2$	$Topic_3$	$Topic_4$	$Topic_5$
2008	0.03	0.07	0.19	0	0.07
2009	0.17	0.08	0.12	0.04	0
2010	0.15	0.05	0.1	0.1	0.1
2011	0.06	0.18	0.06	0.18	0.12
2012	0	0.1	0.3	0.3	0.3

Example. Table 4 shows the topic distributions for five topics of researcher r_i and Table 5 shows the topic distribution after the log weighting scheme has been applied to each year. Eq. (8) shows the topic distribution vector of researcher r_i in year 2009. The vector after inverse log-weight has been applied is depicted in Eq. (9).

$$A^r_{2009k} = [0.4, 0.2, 0.3, 0.1, 0] \tag{8}$$

$$\frac{A^r_{2009k}}{log_2(5)} = [0.17, 0.08, 0.12, 0.04, 0] \tag{9}$$

Furthermore, we adopt a weighted addition of vectors to obtain the final vector as mentioned in Table 5. The final vector A_i for researcher r_i after weighted addition will be:

$$A^r_i = [0.41, 0.48, 0.77, 0.62, 0.59] \tag{10}$$

If we had applied a simple vector addition without any weights, we would have got a vector $A^{r'}_i$ as:

$$A^{r'}_i = [0.9, 0.9, 1.4, 0.9, 0.9] \tag{11}$$

We can clearly see the difference between A^r_i and $A^{r'}_i$. It clearly indicates the influence of topic distribution vector of recent year 2012 in the calculation of A^r_i where as in $A^{r'}_i$, all the year wise vectors contribute equally. Furthermore, researcher–researcher similarity is done among venues exploiting their corresponding weighted vector A^r_i and T^r_i respectively.

6.3. Author2Vec edge weighting

Using A_i and T_i for a researcher r_i, we compute cosine similarity between any two researchers. We get two cosine similarities, $Sim_a(r_i, r_j)$ and $Sim_t(r_i, r_j)$, for a pair of researchers, r_i and r_j, using (A_i, A_j) and (T_i, T_j) respectively.

$$Sim_a(r_i, r_j) = \frac{\boldsymbol{A^r_i} \cdot \boldsymbol{A^r_j}}{|\boldsymbol{A^r_i}||\boldsymbol{A^r_j}|} = \frac{\sum_{b=1}^{k}(a_{b,i} * a_{b,j})}{\sqrt{\sum_{b=1}^{k} a^2_{b,i}} * \sqrt{\sum_{b=1}^{k} a^2_{b,j}}} \tag{12}$$

$$Sim_t(r_i, r_j) = \frac{\boldsymbol{T^r_i} \cdot \boldsymbol{T^r_j}}{|\boldsymbol{T^r_i}||\boldsymbol{T^r_j}|} = \frac{\sum_{b=1}^{k}(t_{b,i} * t_{b,j})}{\sqrt{\sum_{b=1}^{k} t^2_{b,i}} * \sqrt{\sum_{b=1}^{k} t^2_{b,j}}} \tag{13}$$

Now we utilize these two similarity metrics to get one final metric, $Sim(r_i, r_j)$ with the help of an adjustment parameter m as:

$$Sim(r_i, r_j) = m * Sim_a(r_i, r_j) + (1 - m) * Sim_t(r_i, r_j) \tag{14}$$

where $m \in [0, 1]$. We consider these similarity scores as contextual similarity features (CSF). Now, we call this similarity score $Sim(r_i, r_j)$ as $CSF(r_i, r_j)$. The influence of m is discussed in Section 10.5.2. Note that, we can use the above similarity score $CSF(r_i, r_j)$ to create the AAG and also to compute the edge-weight among researchers.

7. The architecture of MRCR model (Layer-3)

The process of the MRCR model mainly consists of four steps: (i) Generation of author–author graph (AAG), (ii) Meta-path features aggregation (MPF), (iii) Scholarly influence-aware features (SIF), and (iv) Recommendation of biased RWR model. We are attempting to discover inherent community structures in an author–author Graph (AAG) to understand the network more profoundly and reveal interesting concepts shared among researchers.

7.1. Generation of author–author graph (AAG)

In this section, we will create a homogeneous undirected author–author graph (AAG) from SIN graph in order to recommend relevant collaborators for a target researcher. We define this graph as an undirected graph $G' = (V', E')$ as defined in Definition 5. AAG is a type of SIN with a node type mapping function δ and an link type mapping function μ as defined in Definition 1. Here, we have one types of vertex V' for each researcher. V'={set of researchers associated with P_main papers}. The type of edge E' is defined as: $v'_1 \xrightarrow{connects} v'_2 : \delta(v'_1), \delta(v'_2) \in \{P\_main\}$, $v'_1, v'_2 \in V'$.

It joins two researchers using only one type of link edge such that $v'_1 \xrightarrow{e'_1} v'_2$, where $\mu(e'_1) \in E'$.

7.1.1. Computation of edge-weight of AAG

After creating the author–author graph (AAG), we need to compute the edge-weight among researchers in AAG. Initially, CSF score $CSF(r_i, r_j)$ as computed in Section 6.3 among researchers (pairwise) is used to create the AAG graph. The average CSF score is used as a threshold to create an edge between researchers. There is no edge that exists with less than average CSF score found among researchers. Initially, this score is used to recommend top N collaborators for a target researcher. This approach is purely content-based so-called TBRec model, and we will use it as a baseline in Section 10.3. Further, we have calculated the combined weighted score $CWS(r_i, r_j)$ between any two researchers r_i and r_j by integrating both meta-path features (MPF) and scholarly influence-aware features (SIF).

7.1.2. Combining different meta-path features into AAG

Since meta-paths are mostly composite relations of various links type in a SIN graph, they can capture the various relationship between SIN nodes [63]. We assume that a meta-path connects two different P_main papers x, y, which are written by two disjoint core researchers r_i, and r_j, respectively.

We observe that meta-path features with more than two degrees are not much meaningful in our work and are not able to create much difference to compute the similarity among researchers. To reduce the time complexity and to obtain a tightly coupled relationship among researchers, only one-degree,[7] and two-degree meta-path features are incorporated into this MRCR model.

[7] The degree of a meta-path indicates its length and the distance between two main papers (P_main).

Please cite this article as: T. Pradhan and S. Pal, A multi-level fusion based decision support system for academic collaborator recommendation, Knowledge-Based Systems (2020) 105784, https://doi.org/10.1016/j.knosys.2020.105784.

Table 6
Meta-paths used in DRACoR model.

No.	Meta-path	Description
1.	*common_author*	Core researcher's share an author (R)
2.	*common_venue*	Core researcher's share a venue (V)
3.	*common_term*	Core researcher's share a term (T)
4.	*direct_cites*	Core researcher cites a paper written by core researcher (P_{main})
5.	*direct_cited_by*	Paper written by a core researcher cited by a core researcher (P_{main})
6.	*citation_paper*	Core researcher's share a reference (P_{ref})
7.	*co_citation_paper*	Papers written by core researcher's co-cited together (P_{cite})

7.2. Meta-path features (MPF)

Table 6 lists all types of meta-paths defined in our model. We are extracting the researcher of *P_main* and considering as a core researcher in order to maintain a homogeneous AAG graph. The following kinds of similarity scores are exploited in order to compute the meta-path features (MPF).

(i) Co-author similarity (C_S)
(ii) Term similarity (T_S)
(iii) Venue similarity (V_S)
(iv) Direct citation based similarity (DC_S)
(v) Co-citation based similarity (CC_S)

The calculation of these above scores is elucidated in the later sections. We use this cumulative score of the above similarity scores to bias the behavior of the random walk such that it will more easily traverse to positive collaborators in the AAG graph. Finally, we generated a list of recommended potential collaborators after the random walk converges.

7.2.1. Computing meta-path edge weights as features

In order to discover the latent association between researchers, we have divided the above seven meta-paths as depicted in Table 6, into the following categories of edge weighting.

(i) Co-author similarity (C_S): We observe that if two researchers have a similar co-authors profile, it makes it easy for researchers to connect. We consider the co-authors profile C_i of a researcher r_i as a vector of length L, equal to the total number of authors. Each dimension of the vector represents an author. The value of C_i at jth index will be 1 if r_i and r_j have worked in the past where r_j represents the author at dimension j and zero if they have not. Index i represents author r_i, and the value at index i for C_i is kept 1.
We calculate the co-author similarity between two authors r_i and r_j by calculating the cosine of the angle between C_i and C_j. Therefore, we can write the co-author similarity $sim_{C_S}(r_i, r_j)$, between r_i and r_j as:

$$sim_{C_S}(r_i, r_j) = \frac{\sum_{l=1}^{L}(C_{i,l} * C_{j,l})}{\sqrt{\sum_{l=1}^{L} C_{i,l}^2} * \sqrt{\sum_{l=1}^{L} C_{j,l}^2}} \tag{15}$$

In reality, none of the similarity scores among two researchers will get a perfect score of 1 and also random walk is sensitive to higher probability score. To avoid such issue, normalization of data within a uniform range (e.g., (0–1)) is essential to prevent larger applies to the output variables. One way is to scale input and output variables (z) in the interval $[\rho_1, \rho_1]$ corresponding to the range of the transfer function [64]. Before adding any individual meta-path score into the model, we are individually applying the normalization to be in the range of [0.1–0.95] as shown in Eq. (16).

$$z_i = \rho_1 + (\rho_2 - \rho_1)\frac{(x_i - x_i^{min})}{(x_i^{max} - x_i^{min})} \tag{16}$$

Algorithm 1: Pseudo-code of MRCR model

```
Input: The AAG Graph with $V' = \{r_1, r_2, \cdots, r_l\}$; a given target
       researcher $r_i$
Output: Top N recommended list of collaborators
Initialize Q based on a given target researcher $r_i$
$R_0 \leftarrow$ Q
Initialize NumIteration
Initialize MinDelta for break
$Sim(r_i, r_j)$, $Avg\_Similarity$=0
for $i \leftarrow 0$ to $|r_l|-1$ do
    for $j \leftarrow 0$ to $|r_l|-1$ do
        if $(i==j)$ then
            $Sim(r_i, r_j) \leftarrow 0$
        else
            Compute $Sim(r_i, r_j)$ by using Eq. (14)
            $Avg\_Similarity \leftarrow Avg\_Similarity + Sim(r_i, r_j)$
        end
    end
end
$Avg\_Similarity \leftarrow \frac{Avg\_Similarity}{|r_l| * (|r_l|-1)}$
for $i \leftarrow 0$ to $|r_l|-1$ do
    for $j \leftarrow 0$ to $|r_l|-1$ do
        if $Sim(r_i, r_j) > Avg\_Similarity$ then
            Create an edge $(r_i, r_j)$ among $r_i$ and $r_j$ in AAG
        else
            Discard the edge $(r_i, r_j)$ from AAG
        end
    end
end
foreach edge $(r_i, r_j)$ in AAG do
    Compute $CWS_{MPF}(r_i, r_j)$ by using Eq. (23)
    Compute $CWS_{SIF}(r_i, r_j)$ by using Eq. (27)
    Compute $CWS(r_i, r_j)$ by using Eq. (28)
end
foreach neighbor $N(r_i)$ of target researcher $r_i$ do
    Compute $w_{r_i,r_j}$(edge weight) by using Eq. (30)
    $S_{i,j}$=$w_{r_i,r_j}$
end
for $k \leftarrow 0$ to $NumIteration - 1$ do
    difference=0
    for $i \leftarrow 0$ to $len(Q) - 1$ do
        $R_{k_i} = \alpha \sum_{j=o}^{len(Q)-1} S_{i,j}R_j$+(1-$\alpha$)$Q_i$
        difference=difference+($R_{k_i}$-$R_{k-1_i}$)
    end
    if $(difference < MinDelta)$ then
        break
    end
end
Sort collaborators in the decreasing order of their ranking scores
Prepare the final list of top N collaborators for $r_i$
```

where z_i is the normalized value of x_i, and x_i^{max} and x_i^{min} are the maximum and minimum values of x_i in the database. After applying this normalization, we will get a normalized $sim_{C_S}(r_i, r_j)$ score $sim'_{C_S}(r_i, r_j)$ among two researchers r_i and r_j.

(ii) Venue similarity (V_S): Venue plays a crucial role in the collaboration of two researchers. When the researchers publish at the same venue, it implies that the research areas are the same. Also, there is a chance that they met at the venue and this might result in their future collaboration with each other. To calculate the venue similarity, we followed a similar approach as Eq. (15). After applying the normalization defined in Eq. (16), we will get a normalized $sim_{V_S}(r_i, r_j)$ score $sim'_{V_S}(r_i, r_j)$ among two researchers r_i and r_j.

(iii) Term similarity (T_S): Term appearing in titles or abstracts of a *P_main* paper after stop word removal and stemming are taken into consideration for this similarity computation. We use snowball stemmer to get the root words [65]. Jaccard similarity coefficient [66] is used to calculate $sim_{T_S}(r_i, r_j)$ (Eq. (17)). Here set E and F denote sample terms occur in all abstracts and titles published by researchers r_i and r_j respectively.

$$sim_{T_S}(r_i, r_j) = \frac{|E \cap F|}{|E \cup F|} \tag{17}$$

where $0 \leq sim_{T_S}(r_i, r_j) \leq 1$. After applying the normalization defined in Eq. (16), we will get a normalized $sim_{T_S}(r_i, r_j)$ score $sim'_{T_S}(r_i, r_j)$ among two researchers r_i and r_j.

(iv) Direct citation based similarity (DC_S): It is experimentally observed that, if two researchers are citing each other very frequently then there is a very high probability that they will work together again. So, we are calculating the number of times they co-cited each other and give the weight-age to the researcher–researcher links. We have used meta-paths such as direct cites and direct cited-by to compute the similarity among researchers. The computation of edge weighting of DC_S is defined below:

$$sim_{DC_S}(r_i, r_j) = Count_1(r_i \rightarrow r_j) + Count_2(r_j \rightarrow r_i) \tag{18}$$

where $Count_1(r_i \rightarrow r_j)$ is the number of times author r_i cites a set of papers written by author r_j and vice versa. After applying the normalization defined in Eq. (16), we will get a normalized $sim_{DC_S}(r_i, r_j)$ score $sim'_{DC_S}(r_i, r_j)$ among two researchers r_i and r_j.

(v) Co-citation based similarity (CC_S): It is experimentally observed that, if two researchers get co-cited by some other paper then there is a very high probability that they can work in the future as their research area might be the same. Similarly, if two researchers are frequently citing common papers, they may work in the future as their research area might be the same. We have used meta-paths features such as co-cites and co-cited-by to compute the similarity among authors. The computation of edge weighting of CC_S is defined below:

$$sim_{CC_S}(r_i, r_j) = Sum_1(r_i, r_j \rightarrow p_i) + Sum_2(p_j \rightarrow r_i, r_j) \tag{19}$$

where $Sum_1(r_i, r_j \rightarrow p_i)$ is the number of times authors r_i and r_j cite a common set of papers. $Sum_2(p_j \rightarrow r_i, r_j)$ is the number of times authors r_i and r_j cited by a common set of papers. After applying the normalization defined in Eq. (16), we will get a normalized $sim_{CC_S}(r_i, r_j)$ score $sim'_{CC_S}(r_i, r_j)$ among two researchers r_i and r_j.

The link weight between any two researchers will be computed through the addition of link weighting scores discussed above. We add each meta-path features into the model and will analyze their effect on the recommendation quality. We already have initial edge weighting score CSF, which is computed by log-weighting based abstract and title similarity as computed in Eq. (14). It was purely based on the contextual similarity. After applying the normalization defined in Eq. (16), we will get a normalized CSF score $CSF'(r_i, r_j)$ among two researchers r_i and r_j. Initially the recommendation will be provided on the basis of this normalized score.

$$CWS(r_i, r_j) = CSF'(r_i, r_j) \tag{20}$$

7.2.2. MPF based combined weighted score $CWS_{MPF}(r_i, r_j)$

We need to combine individual meta-path scores into the model, and we call it as the MPF based combined weighted score $CWS_{MPF}(r_i, r_j)$ as depicted in Eq. (23) to use it as a probability score between researchers in AAG graph as computed using Eq. (30) to apply random walk with restart.

$$CWS_{CF}(r_i, r_j) = sim'_{C_S}(r_i, r_j) + sim'_{T_S}(r_i, r_j) + sim'_{V_S}(r_i, r_j) \tag{21}$$

$$CWS_{OF}(r_i, r_j) = sim'_{DC_S}(r_i, r_j) + sim'_{CC_S}(r_i, r_j) \tag{22}$$

$$CWS_{MPF}(r_i, r_j) = CWS_{CF}(r_i, r_j) + CWS_{OF}(r_i, r_j) \tag{23}$$

All normalized scores obtained from Eq. (15), score obtained from normalized venue similarity (V_S), Eqs. (17)–(19) are added to obtain the meta-path based combined weighted score $CWS_{MPF}(r_i, r_j)$.

7.3. Scholarly influence-aware features (SIF)

To discover the patterns in researcher association over time and to get the latent association between researchers, specifically, we have explored a few scholarly influence-aware features. The following scholarly influence-aware features are taken into consideration.

(i) Percentage of collaboration (PC_S)
(ii) H-Index based level similarity (L_S)

7.3.1. Percentage of collaboration

How frequently a researcher collaborates with another researcher can also indicate the likelihood of collaboration shortly. Moreover, a researcher with whom an author has frequently collaborated can lead to collaboration with other researchers as well. Our idea is that a researcher's frequently collaborated co-authors can play a role in his/her future collaborations.

Let the total number of publications of author r_i be m and the total number of publications of author r_j be n. The number of publications they have in common be p. Then, we define the percentage of collaboration $sim_p(r_i, r_j)$ between r_i and r_j as:

$$sim_{PC_S}(r_i, r_j) = \frac{p}{m} + \frac{p}{n} \tag{24}$$

$$= p(\frac{1}{m} + \frac{1}{n}) \tag{25}$$

Observe that the first term in Eq. (24), represents the fraction of the publications that r_i shares with r_j i.e., p out of all the publications of r_i i.e., m and analogously, the second term represents the fraction of the publications that r_j shares with r_i i.e., p out of all the publications of r_j i.e., n. After applying the normalization defined in Eq. (16), we will get a normalized $sim_{PC_S}(r_i, r_j)$ score $sim'_{PC_S}(r_i, r_j)$ among two researchers r_i and r_j.

7.3.2. H-index based level similarity (L_S)

Normally, researchers having the same academic level imply future collaboration. We have calculated the level similarity among two researchers by using two parameters:

(i) Number of citations
(ii) H-index with individual citations

The level similarity is calculated using the formula:

$$sim_{L_S}(r_i, r_j) = \frac{min(h_i, h_j)}{\sum_{k=0}^{min(h_i,h_j)} \log_2 (|C_{i_k} - C_{j_k}| + 2)} \tag{26}$$

Here L_S represents Level similarity between two authors r_i and r_j having h-index h_i and h_j respectively. The C_{i_k} and C_{j_k} are the citations of Author r_i and Author r_j of k_{th} paper when papers are sorted in decreasing order of their citations. After applying the normalization defined in Eq. (16), we will get a normalized $sim_{L_S}(r_i, r_j)$ score $sim'_{L_S}(r_i, r_j)$ among two researchers r_i and r_j.

7.3.3. SIF based combined weighted score $CWS_{SIF}(r_i, r_j)$

We need to combine scholarly influence-aware features (SIF) based similarity scores into the model, and we call it as SIF based combined weighted score $CWS_{SIF}(r_i, r_j)$ as depicted in Eq. (27) to use it as a probability score between researchers in AAG graph as computed using Eq. (30) to apply random walk with restart.

$$CWS_{SIF}(r_i, r_j) = sim'_{PC_S}(r_i, r_j) + sim'_{L_S}(r_i, r_j) \tag{27}$$

As we mentioned earlier, individual scores mentioned above are normalized before computing the combined weighted score as described in Eq. (16).

7.3.4. Cumulative combined weighted score $CWS(r_i, r_j)$

In addition to normalized CSF score obtained in Eq. (20), $CWS_{MPF}(r_i, r_j)$, and $CWS_{SIF}(r_i, r_j)$ scores are added to obtain the final CWS(r_i, r_j) in order to enhance the probability of recommending relevant collaborators during recommendation. After applying the normalization defined in Eq. (16), we will get a normalized $CWS'_{MPF}(r_i, r_j)$ score and $CWS'_{SIF}(r_i, r_j)$ among two researchers r_i and r_j.

$$CWS(r_i, r_j) = CSF'(r_i, r_j) + CWS'_{MPF}(r_i, r_j) + CWS'_{SIF}(r_i, r_j) \tag{28}$$

7.4. Recommendation of MRCR model

To exploit both collaboration network information along with publication content, we employ a popular network-based approach known as random walk with restart (RWR). The pseudo-code of the MRCR model is given in Algo. 1.

7.4.1. Random walk with restart (RWR)

RWR provides a good way to measure how closely related two nodes are in a graph [67]. The core equation of the RWR model is shown in Eq. (29).

$$R^{(t+1)} = \alpha \mathbf{S} R^{(t)} + (1 - \alpha)Q \tag{29}$$

where $\mathbf{S}$ is the transfer matrix, representing the probability for each node to jump to other nodes. $R^{(t)}$ is the rank score vector at step t and Q is the initial vector of the form $(0, \ldots, 1, \ldots, 0)$.

We use the weighted combined score (CWS) found after aggregating various meta-path features and scholarly influence-aware features in Eq. (28), to bias the walker towards researchers with higher content as well as semantic similarity. Each entries $S_{i,j}$ in S is the transition probability for each researcher r_i in AAG skipping to next researcher r_j. It can be computed as edge weight w_{r_i,r_j} as shown in the equation below:

$$w_{r_i,r_j} = \frac{CWS(r_i, r_j)}{\sum_{r_x \in N(r_i)} CWS(r_i, r_x)} \tag{30}$$

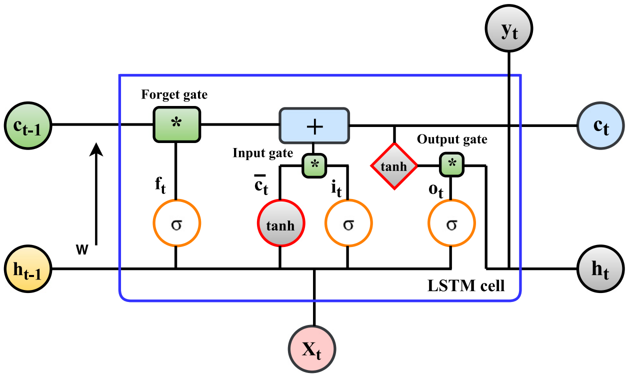


Fig. 3. Basic components of LSTM architecture.

where $N(r_i)$ is set of neighbors who have incoming links from r_i.

Initially, the rank score of the target node is 1, while others are 0. Initially, the vector Q is initialized to $R^{(0)}$, α is the damping coefficient. With probability $(1 - \alpha)$, walker restarts from the start node. RWR is an iterative process. After certain iterations, $R^{(t)}$ converges to a steady-state probability vector. We use $R^{(t+1)}$ researcher-rank score vector to give our final top N recommendation.

8. The architecture of DBCR model (Layer-3)

In this section, we introduce the basics of Recurrent Neural Networks (RNNs) and LSTMs and provide the details of our proposed model based on deep learning to provide a diversified personalized collaborator recommendation.

8.1. Basics of RNNs and LSTMs

Recurrent Neural Networks are the state of the art algorithm for sequential data. In RNN, the information cycles through a loop. It decides by considering the current input and also what it has learned from the inputs it received before. Mathematically, RNNs can be expressed as:

$$L^{(t)} = g(L^{(t-1)}, c^{(t)}, \alpha) \tag{31}$$

where $L^{(t)}$ represents the state of the RNN at timestep t which equals the application of transformation g applied considering the state of RNN at timestep $(t - 1)$, the current input $c^{(t)}$ and the network parameter α which are shared through each timestep t = 1,2, $\cdots$,T. As a result, RNNs takes into account the current input and also what it has learned from the inputs it received earlier, as well. LSTMs enable RNNs to memorize their inputs over a long interval of time. This is because LSTMs contain their information in a memory, that is much like the computer's memory because the LSTM can read, write and delete information from its memory.

The LSTM cell contains the following components:

(i) Forget gate "f"(a neural network with sigmoid)
(ii) Candidate layer "c̄"(a neural network with tanh)
(iii) Input gate "i" (a neural network with sigmoid)
(iv) Output gate "o" (a neural network with sigmoid)
(v) Hidden state "h" (a vector)
(vi) Memory state "c" (a vector)

The equations governing LSTM are as follows:

$$f_t = \sigma(X_t * u_t + h_{t-1} * w_t) \tag{32}$$

$$\bar{c}_t = tanh(X_t * u_c + h_{t-1} * w_c) \tag{33}$$

Please cite this article as: T. Pradhan and S. Pal, A multi-level fusion based decision support system for academic collaborator recommendation, Knowledge-Based Systems (2020) 105784, https://doi.org/10.1016/j.knosys.2020.105784.

$$i_t = \sigma(X_t * u_i + h_{t-1} * w_i) \tag{34}$$

$$o_t = \sigma(X_t * u_o + h_{t-1} * w_o) \tag{35}$$

$$c_t = f_t * c_{t-1} + i_t * \bar{c}_t \tag{36}$$

$$h_t = o_t * tanh(c_t) \tag{37}$$

where X_t is the input vector, h_{t-1} is the previous cell output, c_{t-1} is previous cell memory, h_t is current cell output, c_t is current cell memory, $\{w, u\}$ denotes weight vectors for forget gate(f), candidate(c), input gate(i) and output gate(o) respectively. Fig. 3 describes the computational graph of LSTM at time step t.

In this article, we will show how LSTMs will be used to learn embeddings for the textual content describing the items recommended by the content based recommender system. The optimization objective of the skip-gram model is to maximize the following log-likelihood function:

$$L = \sum_{w \epsilon C} logP(context(w)|w) \tag{38}$$

The key point is to construct and calculate the conditional probability function $P(context(w)|w)$. For skip-gram model, given the central word w, we need to predict the words in context(w). Most of the parameter for Word2Vec is taken as default value except for the vector size which is set to be a relatively larger value so that the proposed model can take the entire sentence as context while training a word of the sentence.

8.2. Label selection

In the case of a collaborator recommender system, there are no user ratings, unlike other content-based recommender systems. In a supervised deep learning-based recommender system, we have to give the label so that the model can learn the training parameters. To achieve this, we have taken various parameters by history among researchers. The features used here are similar to the features adopted in the MRCR model except for content similarity, which is as follows:

(i) Venue similarity (V_S)
(ii) Direct citation based similarity (DC_S)
(iii) Co-citation based similarity (CC_S)
(iv) Percentage of collaboration(PC_S)
(v) H-Index based level similarity(L_S)

Calculation of these above mentioned features has been shown in Section 7.2.1. Individual scores mentioned above are normalized before computing the $S(r_i, r_j)$ by Eq. (16). So this normalized $S(r_i, r_j)$ is considered as the label for the DBCR model.

$$S(r_i, r_j) = V_S + DC_S + CC_S + PC_S + L_S \tag{39}$$

8.3. Proposed architecture

The architecture of DBCR model is shown in Fig. 4. This architecture can predict a score S(r_i, r_j) to define the probability that the pair of authors r_i and r_j will collaborate in the future. Briefly, this approach is based on two different word embeddings for two different authors.

This word embeddings can jointly learn continuous vector representations for a pair of authors $r_i \in R$ and $r_j \in R$ that are used to feed a classifier which generates the score which is a probability of their future collaboration. Overall, the proposed architecture consisting of the following six layers:

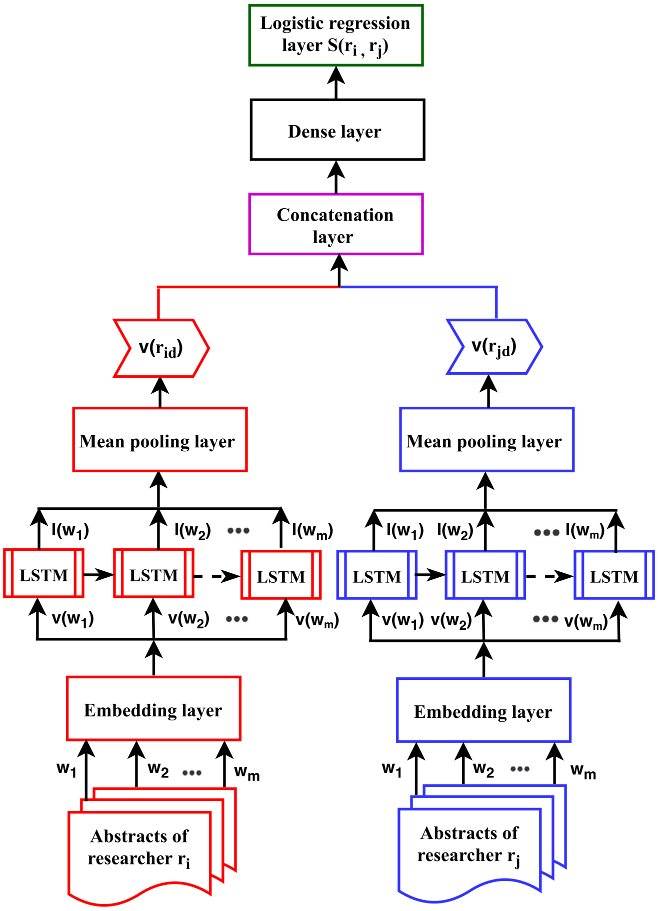


Fig. 4. The architecture of DBCR model.

(i) **Embedding layer**: Generates the matrix associated with author's content from trained Word2Vec model.
(ii) **LSTM layer**: An RNN network with LSTM units.
(iii) **Mean pooling layer**: It down-samples the input by dividing it into rectangular pooling regions and computing the mean values of each region.
(iv) **Concatenation layer**: Concatenates the input vectors.
(v) **Dense layer**: Deep Neural network with hidden layers.
(vi) **Logistic regression layer**: Exploits logistic regression to calculate the score for the pair of authors.

8.3.1. Embedding layer

An important component of DBCR model is the embedding layer, which generates the dense representations of its input. This is our input layer through which abstracts $w_1, w_2, w_3, \dots, w_m$ of the papers written by authors are passed. It will generate the matrix associated with the author's content from the trained Word2Vec model. Given a set of elements R, each of them can be represented as an x-dimensional vector contained in a $E \in \mathbb{R}^{|R| \times x}$ embedding matrix generated using Word2Vec. Each pair of authors is given in input to an embedding layer, which generates an x-dimensional author embedding E. Finally, we will get the new embeddings as $v(w_1), v(w_2), v(w_3), \dots, v(w_m)$.

8.3.2. LSTM layer

This layer contains a RNN network with a few LSTM units. After getting the word embedding matrix, each word representations are sequentially passed through a Long Short Term Memory(LSTM) network with t hidden units which generate for each

Please cite this article as: T. Pradhan and S. Pal, A multi-level fusion based decision support system for academic collaborator recommendation, Knowledge-Based Systems (2020) 105784, https://doi.org/10.1016/j.knosys.2020.105784.

of them a t-dimensional latent representation L using a LSTM cell. The motivation behind the selection of LSTM network are as follows:

(i) As stated in the introduction, several works already showed that LSTM could overcome shallow models.

(ii) Moreover, we chose LSTM since such networks are very effective when sequences of input have to be shaped.

Given that the textual description of the input can be easily viewed as a sequence of words, it was straightforward for us to investigate the adoption of LSTMs in a content-based recommendation scenario. In this LSTM layer, we obtained the latent representations l(w_m) for each embedding $v(w_m)$. We have used tanh as activation function in lstm. The tanh activation function is defined as:

$$f(x) = tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{40}$$

We have used sigmoid as recurrent activation in LSTM. The sigmoid activation function is defined as:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{41}$$

8.3.3. Mean pooling layer

In recurrent-neural-network-based models, pooling is often used to aggregate hidden states at different time steps (i.e., words in a sentence) to obtain sentence embedding. It is used to compress our features to lower fidelity. A mean-pool layer compresses by taking the mean activation in a block. It calculates the mean of the input vectors. After the latent representation are computed by the LSTM network, the pooled vector $v(r_{id})$ (r_{id} denotes researcher r_i) is obtained by a mean pooling layer which averages the latent representations $l(w_m)$ for all the words in the textual description of the item.

The pooling layer has two hyperparameters, the spatial extent of the filter f and the stride s. It takes M, an input volume of size m × x (x is the size of the vector generated by Word2Vec) and provides an output volume of size $\bar{m} \times \bar{x}$ (where $\bar{m} = \frac{m-f}{s+1}$, and $\bar{x} = \frac{x-f}{s+1}$). The pooling layer operates by defining a window of size f*f and reducing the data within this window to a single value which is the average of all values in case of mean pooling layer. The window is moved by s positions after each operation, and the reduction is repeated at each position of the window until the entire activation volume is spatially reduced.

8.3.4. Concatenation layer

This layer mainly concatenates the input vectors resulting after mean pooling layer. We have specifically used this layer to compare them using a deep neural network. The resulting vectors $v(r_i)$ and $v(r_j)$ of the pair of authors respectively are then concatenated through a concatenation layer.

8.3.5. Dense layer

Dense layer is used to compare the feature vectors after Concatenation. Also, the dimensionality of this output until now does not equal to the dimensionality of the desired target. This layer consists of deep neural network with hidden layers. The resulting ($t'_i + t'_j$)-dimensional (t'_i is the reduced t_i-dimensional vector) feature vector is given as input to the dense layers to generate the functions to find the relation between them. We have used sigmoid as activation in the dense layer.

8.3.6. Logistic regression layer

Exploits logistic regression to calculate the score for the pair of authors. Finally, it is passed through the logistic regression layer to predict the score S(r_i, r_j). Mathematically this can be expressed as:

$$S(r_i, r_j) = sigmoid(W_{ih}[v(r_i), v(r_j)] + b_{ih}) \tag{42}$$

where $W_{ih} \in \mathbb{R}^{(t'_i+t'_j)\times 1}$ is a weight matrix, $b_{ih} \in \mathbb{R}$ is a bias term and the square bracket denotes the concatenation operation between two vectors.

The logistic regression layer can learn its parameters W_{ih} and b_{ih} according to the relationship between the authors r_i and, r_j. To generate the top-N recommendations for author r_i, the recommender system generates a list for an author with all other authors sorted in descending order of score S(r_i, r_j). We have used mean squared error as loss function. The mean squared error is defined as:

$$\text{Mean squared error(MSE)} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \bar{Y}_i)^2 \tag{43}$$

where Y_i is the observed value and $\bar{Y}_i$ is the predicted value.

Algorithm 2: Fusion of MRCR and DBCR models

```
Input: Given target researcher $r_i$ for the recommendation
Output: Top N recommended list of collaborators
Initialization
let R= $r_1$, $r_2$...,$r_N$ be the set of target researchers
Perform MRCR model at target author $r_i$ in order to Compute top N similar collaborators
$\mathcal{U}_i$ = Ordered list of unique collaborators found by MRCR model (Section 7)
= $\{a_1, a_2, \ldots, a_N\}$
Perform DBCR in order to Compute top N similar collaborators
$\mathcal{V}_i$ = Ordered list of unique collaborators found by MRCR model (Section 8)
= $\{b_1, b_2, \ldots, b_N\}$
for $i \leftarrow 0$ to $|a_N|-1$ do
    Borda Count $B_c(a_i) \leftarrow N - i + 1$
end
for $j \leftarrow 0$ to $|b_N|-1$ do
    Borda Count $B_c(b_j) \leftarrow N - j + 1$
end
k=0, $counter(a_i)$=false, $counter(b_j)$=false
for $i \leftarrow 0$ to $|\mathcal{U}_i|$-1 do
    for $j \leftarrow 0$ to $|\mathcal{V}_i|$-1 do
        if ($a_i == b_j$) then
            Boda Count $B_c(v_k) \leftarrow B_c(a_i) + B_c(b_j)$
            /* same collaborator so add their Borda Count */
            k=k+1
            $counter(b_j)$=true
        else
            $B_c(v_{k+1}) \leftarrow B_c(a_i)$
            /*individually consider Borda Count of $a_i$*/
            k=k+1
        end
    end
end
for $j \leftarrow 0$ to $|\mathcal{V}_i|$-1 do
    if ($counter(b_j)$! = $true$) then
        $B_c(v_{k+1}) \leftarrow B_c(b_j)$
        /*individually consider Borda Count of $b_j$*/
        k=k+1
    end
end
Sort collaborators in the decreasing order of Boda count $B_c(v_k)$
Prepare the final list of top N collaborators recommendation
```

Please cite this article as: T. Pradhan and S. Pal, A multi-level fusion based decision support system for academic collaborator recommendation, Knowledge-Based Systems (2020) 105784, https://doi.org/10.1016/j.knosys.2020.105784.

9. Fusion model: DRACoR (Layer-4)

The main assumption of fusion-based approach can be stated that "hybrid recommendation approaches can provide more accurate recommendation than a single approach and the disadvantages of one approach can be overcome by the other approach" [10,68]. On the other hand, hybrid approach is a promising alternative to traditional approach. It has shown excellent performance in the field of recommendation [10,69–71]. Data combination has also been widely investigated in the recommendation community. They were often divided into two categories: score-based and ranking-based [72,73]. Ranking-based combination methods require rank or position information to integrate different candidates ranking lists, such as Borda fusion, Condorcet fusion, and MAPFuse.

The MRCR and DBCR models which have been improved with previous publication content, meta-paths features aggregation, random walk with restart, LSTM based deep learning method are integrated into a fusion model based collaborator recommendation approach, i.e., DRACoR. We employed a rank-based fusion technique Borda Count to integrate the existing prediction lists generated by the MRCR model and DBCR model respectively. To be more specific, the predictions resulting from the MRCR and DBCR are firstly produced separately with the purpose of allowing us to leverage the individual strengths of both approaches since there is no interdependency between them, then we are fusing the results with standardized Borda Count technique as mentioned in Algo. 2.

10. Experiments

In this section, we present the experiments of the proposed fusion model DRACoR, where initial two sections present the experimental datasets and evaluation metrics. Then the baseline methods, experimental setting, and parameter tuning are described in further section. The following experiments are performed on a laptop with 64-bit windows 10 operating system, Intel i7-3540M CPU@3.00 GHz, and 32-GB memory. All the programs are implemented in Python and we use a TITAN XP graphic card for learning.

10.1. Data collection

We use two real-world datasets to demonstrate the effectiveness of our proposed method. The first dataset is DBLP-citation-network V10,[8] the citation data extracted from DBLP, ACM, MAG (Microsoft Academic Graph), and other sources [74]. The tenth version contains 3,079,007 papers and 25,166,994 citations. After data preprocessing, we shortlisted 5,072 concurrent authors from 2,408,010 publications. Next, a modified heterogeneous network containing 5,084 terms, 32,018 cited papers, and 93 journals were constructed. We divided the dataset into two parts according to the year of publication: data before the year 2012 as a training set, and the rest as a testing set.

The second dataset was hep-th (Theoretical High Energy Particle Physics) provided by KDD Cup 2003.[9] After data preprocessing as above, we get 1,922 concurrent authors from 20,961 publications. Next, a modified heterogeneous network containing 2,395 terms, 12,018 cited papers, and 64 journals are constructed. We divided the dataset into two parts according to the year of publication: data before the year 1999 as a training set and the rest as a testing set.

10.2. Evaluation metrics

We employed various evaluation metrics such as Precision, Recall, F1-score, nDCG, and MRR, which are quite popular in recommender systems to demonstrate the effectiveness of DRACoR.

(a) Precision, Recall and F1-score: We can divide all nodes (researchers) into four groups according to the following four cases:

- A: collaborating with the target node and recommended;
- B: collaborating with the target node but not recommended;
- C: not collaborating with the target node but recommended;
- D: not collaborating with the target node and not recommended.

$$Precision = \frac{|A|}{|A+C|} \tag{44}$$

$$Recall = \frac{|A|}{|A+B|} \tag{45}$$

$$F1 = \frac{2(Precision * Recall)}{Precision + Recall} \tag{46}$$

(b) Normalized discounted cumulative gain (nDCG): It represents the ratio of discounted system gain and discounted ideal gain accumulated at a particular rank p, where gain at a rank p is the sum of relevance values from rank 1 to rank p [75]. Relevance value (rel_p) is the relevance score (0 or 1) assigned to each recommended collaborators based on the original collaborations (Ground truth data) at position j. Ideal vector is constructed hypothetically where all relevance scores (rel_{ij}) are ordered in decreasing order to ensure the highest gain at any rank.

$$DCG_{sp} = rel_{s1} + \sum_{j=2}^{p} \frac{rel_{sj}}{log_2(j)} \tag{47}$$

$$IDCG_p = rel_{i1} + \sum_{j=2}^{p} \frac{rel_{ij}}{log_2(j)} \tag{48}$$

$$nDCG_p = \frac{DCG_p}{IDCG_p} \tag{49}$$

(c) Mean Reciprocal Rank (MRR): MRR is the arithmetic mean of reciprocal rank (RR) which is the inverse of the first rank where the correct collaborator is recommended in the ranked result [75].

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_{rel_i}} \tag{50}$$

where $rank_{rel_i}$ denotes the rank position of the first relevant collaborator for the ith recommended collaborator in a given target researcher set Q.

10.3. Baseline methods

To measure effectiveness of the proposed system DRACoR, we compare our results with following state-of-the-art methods as discussed below.

[8] https://aminer.org/citation.

[9] https://www.cs.cornell.edu/projects/kddcup/datasets.html.

Please cite this article as: T. Pradhan and S. Pal, A multi-level fusion based decision support system for academic collaborator recommendation, Knowledge-Based Systems (2020) 105784, https://doi.org/10.1016/j.knosys.2020.105784.

(a) CNRec: It is a common neighbors based recommendation model which is quite popular in recommendation based on social-networks. It is based on the assumption that if two researchers have a lot of co-authors in common, there is a high probability that they may collaborate in future [37].

(b) RWR: It involves a basic random walk with restart model on the whole co-author network to recommend collaborators. This model is similar to ACRec, but the probability of skipping to the next neighbor nodes are equal in RWR [34].

(c) TBRec: This model is also called content-based model, which uses LDA on abstract and Doc2Vec on title in order to define the feature vector. This model is also a part of DRACoR without the inclusion of any other factors. We compute the content similarity among researchers by using Eq. (14) to recommend personalized collaborators.

(d) MVCWalker: It is based on RWR-based recommendation, which uses three academic factors: co-author order, latest collaboration time, and times of collaboration within a co-author graph, to recommend personalized collaborators [35].

(e) CCRec: This model exploited both content as well as social network approach in order to recommend collaborators. They incorporated Word2Vec to identify the academic domains, as well as a random walk model to compute researchers' feature vectors [33].

(f) BCR: This model utilized an integrated approach of three academic features: topic distribution of research interest, interest variation with time and researchers impact in collaborator network to provide beneficial collaborator recommendation [29].

(g) RWR-CR: This approach uses a heterogeneous bibliographic network with multiple types of nodes and links with a simplified network structure by removing the citing paper nodes. To weight edges in the network, both sequence importance and freshness importance are taken in to consideration in order to bias the random walker's behaviors [28].

10.4. Experimental setting

While preparing the test dataset, we considered two scenarios: firstly, due to operational constraints, 14 sub-domains of computer science: information retrieval, image processing, security, wireless sensor network, machine learning, software engineering, computer vision, artificial intelligence, data mining, algorithms and theory, databases, natural language processing, parallel and distributed systems, and multimedia were selected as the testing dataset in our experiment.

Secondly, while identifying the target researchers, the following conditions are taken into consideration to measure the effectiveness of DRACoR to handle cold start issues like a new researcher or researcher with less number of publications or collaborations. To validate the effectiveness of DRACoR against new researcher or researchers with fewer collaborations, primarily the below two categories are taken into consideration. There are two major categories, i.e., (a) number of citations (n_c), and (b) target nodes degree (n_d). Generally, n_c denotes the number of citations of individual researchers and n_d denotes the number of collaborators or degree of target researchers. The following two categories are discussed below.

(a) Target researcher's academic level (Number of citations)

(i) *Primary Level* ($2 \leq n_c < 6$)
(ii) *Intermediate Level* ($6 \leq n_c < 26$)
(iii) *Advanced Level* ($26 \leq n_c$)

(b) Target researcher's degree (Number of collaborations)

(i) *Group I* ($1 \leq n_d < 10$)
(ii) *Group II* ($10 \leq n_d < 19$)
(iii) *Group III* ($19 \leq n_d < 29$)
(iv) *Group IV* ($29 \leq n_d$)

Table 7
Experimental parameter settings.

Parameter	Range	Default
Vector dimension (A_i and T_i)	(10-200)	100
Adjustment parameter (m)	(0.1–0.95)	0.7
Damping constant (α)	(0.1–0.95)	0.8
Target researcher's academic level (n_c)	≥ 0	(6–24)
Target researcher's degree (n_d)	≥ 0	≥ 30
Number of iteration	(10–100)	25
Number of recommended nodes	(5–150)	120

In this experiment, the relevance value r is binary, i.e., r ∈ {0 or 1}. It is set to 1 if the recommended collaborators are matching with the ground truth data and set to 0 if the recommended collaborators are not collaborating with the target node in reality.

To comprehensively evaluate our proposed method, we prefer to examine the following research questions (RQs) about various existing issues.

RQ1: How does different parameter selection affect the performance of DRACoR?
RQ2: How does DRACoR handle the cold-start issue for new researchers?
RQ3: How effective is DRACoR in comparison to other state-of-the-art methods?

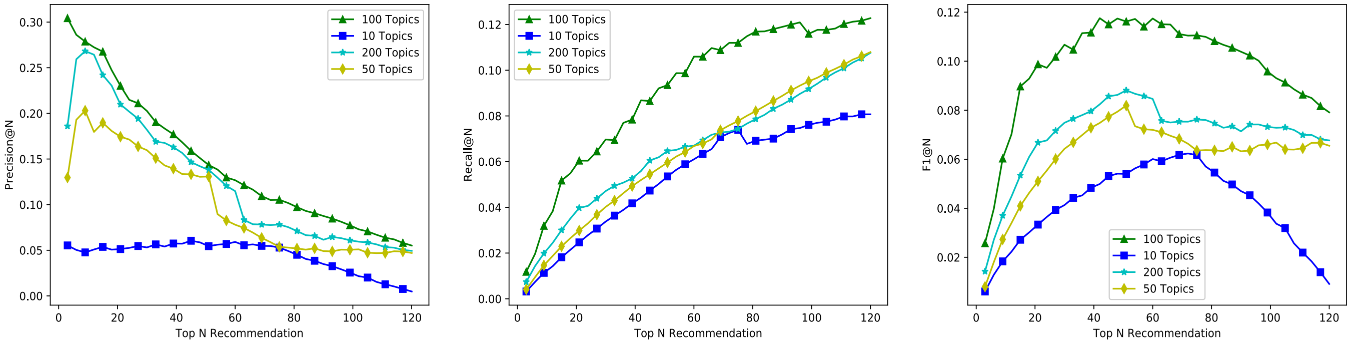
10.5. Parameter tuning and optimization

In this section, we demonstrate the impact of various experimental parameter settings, including vector dimension (A_i and T_i), adjustment parameter (m), damping constant (α), target researcher's academic level (n_c), target researcher's degree (n_d), and number of iteration. The ranges and default values of the parameters are depicted in Table 7. When the effect of the parameter is under examination, the other parameters are set to default values. During this experimentations, the best result and second-best performer are marked by the 'bold-face' and + marks sign in each position.

10.5.1. Influence of vector dimension

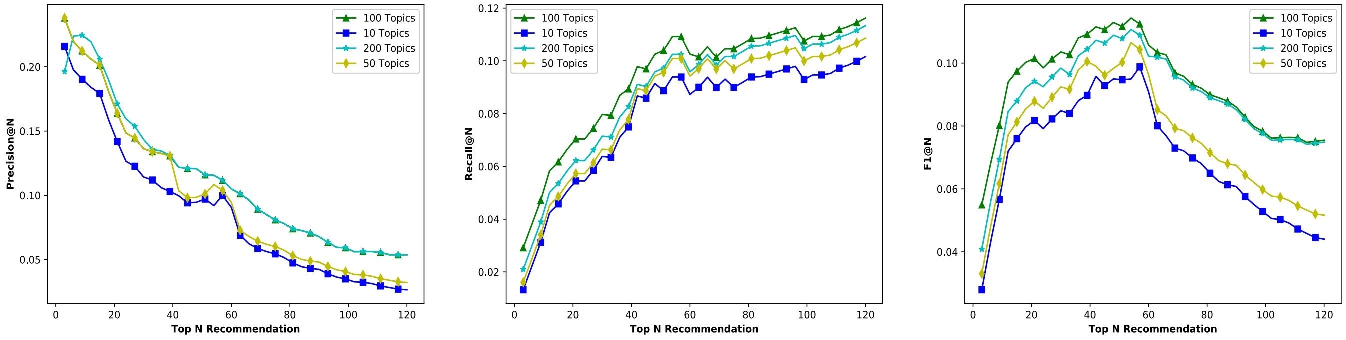
In order to find the ideal dimension for vectors A_i and T_i, we conduct experiments on four values for vector dimension , i.e. {10,50,100,200}. The value of the adjustment parameter is set to be 0.7, and α is set to be 0.8. We choose 140 researchers randomly as the target nodes and run DRACoR model for both the datasets of DBLP and hep-th. This is done to calculate the average precision, recall, and $F1$ over these recommended collaborators. We conducted extensive experiments with different recommendation lists in length to evaluate the influence of the vector dimension on the result. Figs. 5(a) and 6(a) show the performance of our model in terms of precision for different vector dimensions. Similarly Figs. 5(b), 6(b) and 5(c), 6(c), demonstrate the effectiveness of DRACoR in terms of both recall and F1 respectively.

During the experiments on DBLP and hep-th datasets, it can be seen that the model performs best, in terms of precision, when the value of the vector dimension is 100 and performs a downtrend with the recommendation list increasing. In the case of recall evaluation, the overall results show an upward trend and then slightly flattens out at the end of the recommendation list. The best performance of recall is achieved with a vector



(a) Influence of vector dimensions on precision (b) Influence of vector dimensions on recall (c) Influence of vector dimensions on F1

Fig. 5. Influence of vector dimensions on precision, recall and F1 (hep-th).



(a) Influence of vector dimensions on precision (b) Influence of vector dimensions recall (c) Influence of vector dimensions on F1

Fig. 6. Influence of vector dimensions on precision, recall, and F1 (DBLP).

Table 8
Influence of adjustment parameter on MRR.

Adjustment	MRR						
prob.(1-m)	2<=n_c <6	6<=n_c <25	26<=n_c	2<=n_d <10	10<=n_d <20	20<=n_d <30	30<=n_d
0.5	0.0793	0.0849	0.0853	0.0854	0.0861	0.0864	0.0895
0.45	0.0798	0.0879	0.0851	0.0853	0.0893	0.0847	0.0853
0.4	0.0867	0.0905	0.0893	0.0915	0.0841	0.0859	0.0874
0.35	0.0972	0.0895	0.0949	0.0858	0.0903	0.0885	0.0896
0.3	**0.1093**	**0.1197**	**0.1267**	**0.1134**	**0.1127**	**0.1185**	**0.1189**
0.25	0.0976^+	0.1014^+	0.1258^+	0.1016^+	0.1039^+	0.1073^+	0.1052^+
0.2	0.0668	0.0848	0.1132	0.0894	0.0917	0.0995	0.0987
0.15	0.0526	0.0773	0.0866	0.0739	0.0877	0.0914	0.0923
0.1	0.0473	0.0637	0.0725	0.0683	0.0746	0.0828	0.0848
0.05	0.0437	0.0591	0.0683	0.0565	0.0677	0.0769	0.0787

dimension of 100. The F1 score performs the upper convex curve, rapidly rising and then shows a slight decline. The best performance of F1 score is achieved with a vector dimension of 100. So considering the above performance, in this experiment, the value of the vector dimension has been taken as 100.

10.5.2. Influence of an adjustment parameter (m)

This parameter has a realistic significance as it controls how an abstract and a title of research papers published by a researcher determines the area of interest of a researcher. In this section, we analyze how the adjustment parameter (m) influences the performance of the algorithm concerning nDCG and MRR. In order to find the ideal value of m to get the efficient combined score of vectors A_i and T_i, we conducted experiments on 10 possible values for adjustment parameter, i.e. {0.5, 0.45, 0.4, 0.35, 0.3, 0.25, 0.2, 0.15, 0.1, 0.05 }. The value of vector dimension (A_i and T_i) is set to be 100 and α is set to be 0.8.

From Table 8, we can observe that the variation tendency of MRR score performs roughly consistent. We can see that the MRR shows an overall upward trend with the increasing value of adjustment parameter m of value 0.7. The model performs the best while the value of the adjustment parameter (m) is 0.7 as marked in bold text. This is because, in most of the cases, the abstract is giving a better clarity of topic similarity while in a few cases the title is resulting better. So considering this experiment in a similar nature, the value of (1-m) has been taken as 0.3.

10.5.3. Influence of damping constant (α)

This parameter has a realistic significance as it controls how far the random walker reaches. In this section, we analyze how the damping coefficient influences the performance of the algorithm concerning nDCG and MRR. With higher values of α, the probability of random walker reaching far away nodes increases. Hence, the number of new collaborators, i.e., researchers who have not collaborated with the target researcher in training set but have done so in the test set, increases. It is evident from Table 9, that as the damping constant increases there is a drastic

Table 9
Influence of restart probability on nDCG and MRR.

Restart prob. (1-α)	nDCG (New)	nDCG (O)	MRR(N)	MRR(O)
0.5	0.009	0.338	0.005	0.419
0.45	0.010	0.337	0.023	0.407
0.4	0.017	0.339	0.046	0.418
0.35	0.026	0.339	0.051	0.418
0.3	0.036	0.343	0.078	0.426
0.25	0.104	0.348^{+}	0.084	0.428^{+}
0.2	**0.162**	**0.419**	**0.179**	**0.494**
0.15	0.107^{+}	0.297	0.127^{+}	0.417
0.1	0.089	0.254	0.123	0.329
0.05	0.061	0.226	0.119	0.121

increase in MRR and nDCG for new collaborators and a negligible decrease in MRR and nDCG for overall (new+old) collaborators.

The table displays the influence of restart probability $(1-\alpha)$ on the algorithm. This parameter setting gives the highest nDCG of 0.162 and 0.419 for both new and old collaborators. Similarly, while evaluating MRR, we can see that the overall results of MRR for both new and old collaborators are 0.179 and 0.494 respectively. The second-best performer is indicated with a $+$ marks sign. Considering the above results of both nDCG and MRR for new and old collaborators, in this experiment, the value of (1-α) has been taken as 0.2.

10.5.4. Influence of target researcher's academic level

In this section, we demonstrate the overall performance of the DRACoR model against varying academic levels of target researchers. The experimental settings are the same as other groups of experiments. The vector dimension is 100, the adjustment parameter is 0.7, and the damping constant was 0.8 during the experiment.

In Figs. 7(a) and 8(a), the precision of DRACoR performs better on recommending potential collaborators for intermediate and advanced level researchers on both hep-th and DBLP datasets. For the primary level researchers, it shows a relatively low precision value. However, according to Figs. 7(b) and 8(b), DRACoR is good at recommending for those primary level researchers in terms of recall. However, the performance of DRACoR shows the worse for advanced-level researchers.

DRACoR shows higher F1 score on recommending for the intermediate level researchers compared to the primary and advanced-level researchers (Figs. 7(c) and 8(c)). After seeing all the analysis of the results, we observe that the academic level of target researchers has a great impact on the performance of DRACoR. If we focus more on the overall metric F1 rating, the DRACoR is higher at recommending potential collaborators for the one's intermediate level researchers.

10.5.5. Influence of target researcher's degree

The complete results are shown in Figs. 9 and 10 to validate the influence of target node's degree. In terms of precision, the larger the target node's degree, the better the model's performance (Figs. 9(a) and 10(a)). Besides, we can see that DRACoR has relatively higher precision with group IV than all other groups. At the range from 0 to 10, DRACoR performs the worst. But when the target node's degree gets larger than 30, the precision performs better as compared to other groups of target researchers. Thus we can conclude that DRACoR has higher precision for strong nodes but performs almost the same for weak nodes.

Figs. 9(b) and 10(b) show the comparison of recall rate with the changing degree. Similar to the results of precision, when the degree becomes larger than 30, the corresponding recall rate of DRACoR increases. Besides, we can see that DRACoR has a relatively higher recall with group IV than all other groups.

Figs. 9(c) and 10(c) show the comparison of F1 with the changing degree. But when the target node's degree gets larger than 30, the F1 performs better as compared to other groups of target researchers. Thus we can conclude that DRACoR has a higher recall for strong nodes but performs almost the same for weak nodes.

10.5.6. Impact of number of iteration on overall results

In this paper, the higher the number of iterations, the higher the number of matrix multiplication operations done by RWR before getting the recommended list. While evaluating the overall performance of DRACoR, it has been observed that, there is no significant changes occurring when iteration times get bigger. As shown in Figs. 11(a) and 12(a), the model achieves a maximum precision of 16% and 11% at iteration 21 and 23 in both hep-th and DBLP respectively. There are similar behavior observed by the model in case of recall and $F1$ until 23 iterations as shown in Figs. 11(b), 12(b), 11(c) and 12(c). Afterward, the model becomes convergent. So there is no need to execute the model with many iterations. Based on the above experiments, we have set the iteration time as 25.

10.6. Parameter tuning of DBCR model

The model is trained using RMSProp as an optimizer. The parameter α is set to be 0.9, and the learning rate is set to 0.0001. Models are trained for 50 epochs, by setting batch sizes to 512. As for cost function, we choose the mean squared error, which is typically used for regression tasks since it tries to minimize the mean squared error in the regression. Due to the computational costs requested by the models, the dimension of the learned embeddings r_i and r_j are fixed to 50.

11. Results and discussions

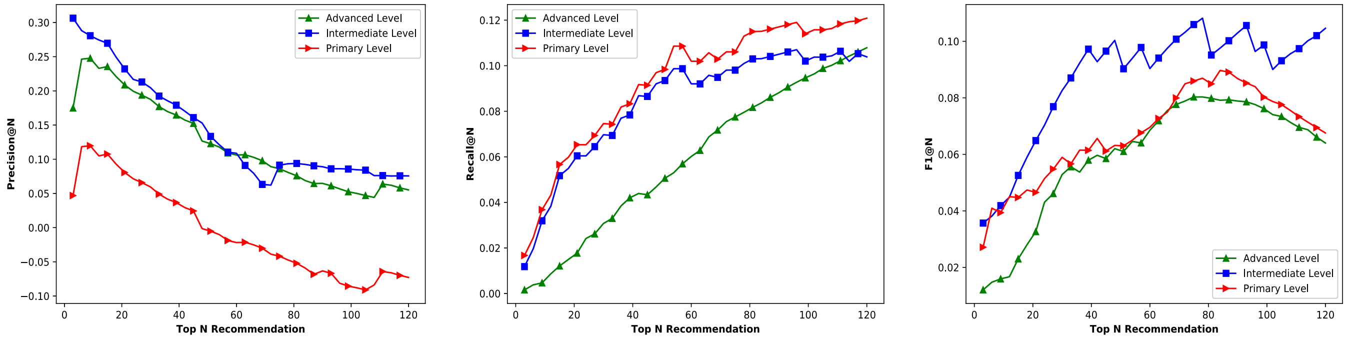
In this section, we evaluated the effectiveness of DRACoR against existing state-of-the-art methods. Before evaluating the performance of the fusion model, DRACoR individual performance analysis of MRCR and DBCR models are estimated.

11.1. Results analysis of MRCR model

The detailed results are shown in Figs. 13 and 14 respectively. While evaluating on the hep-th dataset, the MRCR model achieves the highest precision of 0.304 at first recommendation, and slowly, it shows a downward trend and reaches a precision value of 0.055 at position 120 as shown in Fig. 13(a). Similarly, the MRCR model achieves the highest precision of 0.224 at first recommendation, and slowly it shows a downward trend and reaches a precision value of 0.028 at position 120 on DBLP dataset, as shown in Fig. 14(a).

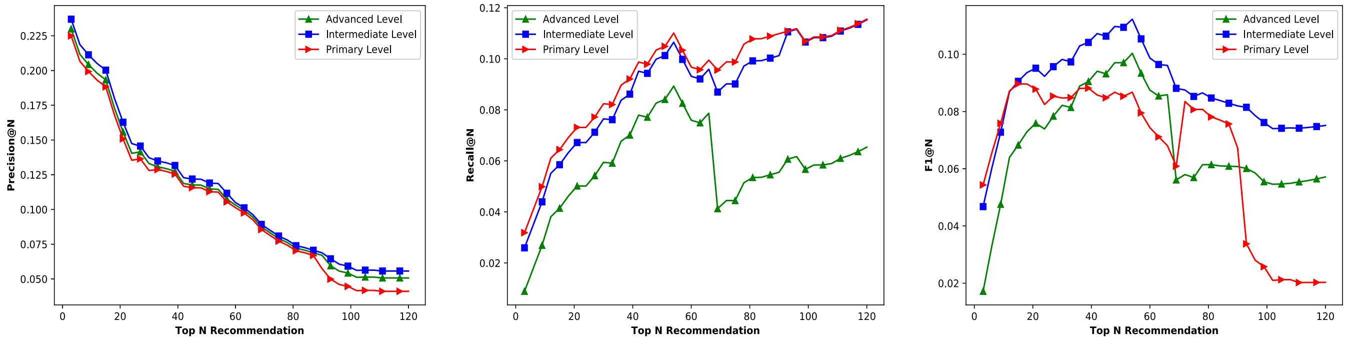
In the case of recall evaluation on hep-th, the MRCR model performs an upward trend and reaches the highest recall of 0.098 at position 54, and afterward again it shows a downward trend and reaches a recall of 0.075 at position 78. Then it slowly increases and achieves the highest recall of 0.098 at position 120, as shown in Fig. 13(b). It provides a similar nature of performance on DBLP dataset too. As shown in Fig. 14(b), it recommends with a higher recall of 0.098 at position 56, and afterward, it seems very like a trend of decline on recall to a certain degree and reaches a recall of 0.096 at position 120.

Similarly, while evaluating the performance on hep-th dataset, the MRCR model shows an upward trend from the beginning and reaches a F1 of 0.113 at position 50, and slowly it shows a downward trend and finally achieves a F1 of 0.0793 at position 120 as shown in Fig. 13(c). But in case of DBLP dataset, the MRCR model



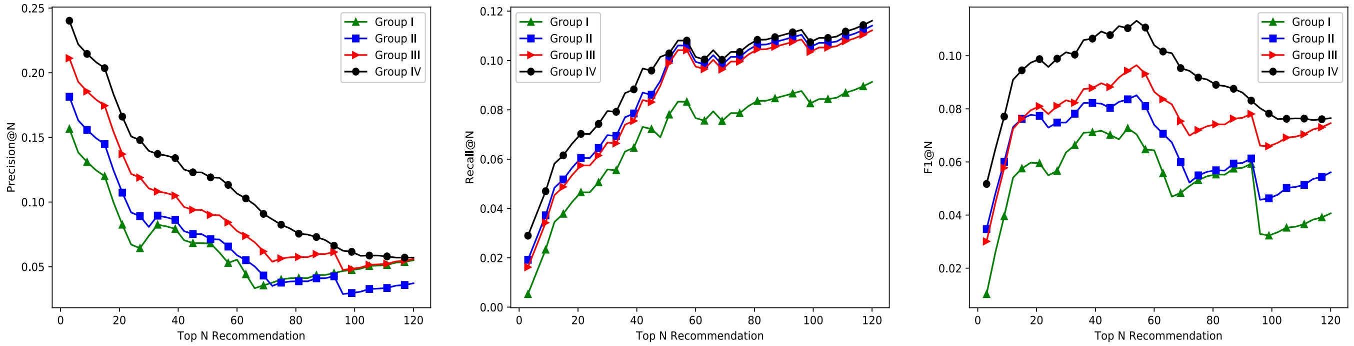
(a) Influence of academic level on preision (b) Influence of academic level on recall (c) Influence of academic level on F1

Fig. 7. Influence of researcher's academic level on precision, recall and F1 (hep-th).



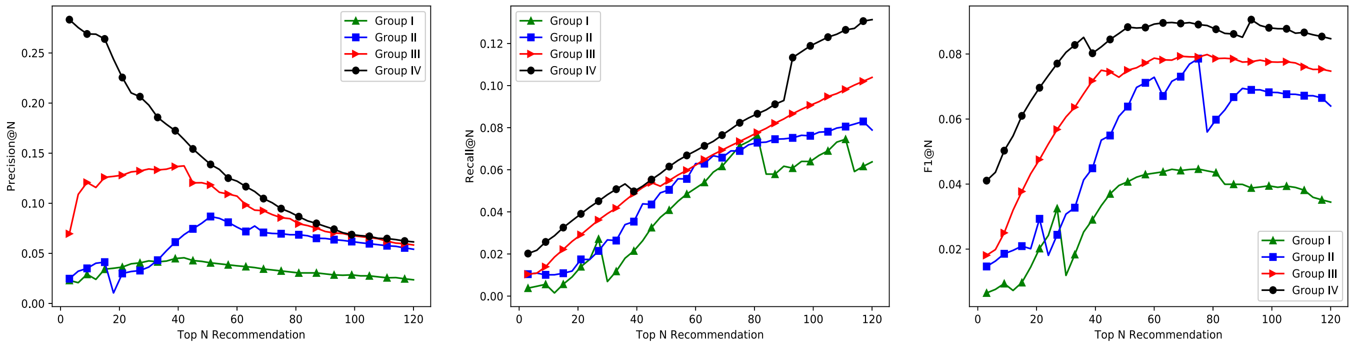
(a) Influence of academic level on preision (b) Influence of academic level on recall (c) Influence of academic level on F1

Fig. 8. Influence of researchers academic level on precision, recall, and F1 (DBLP).



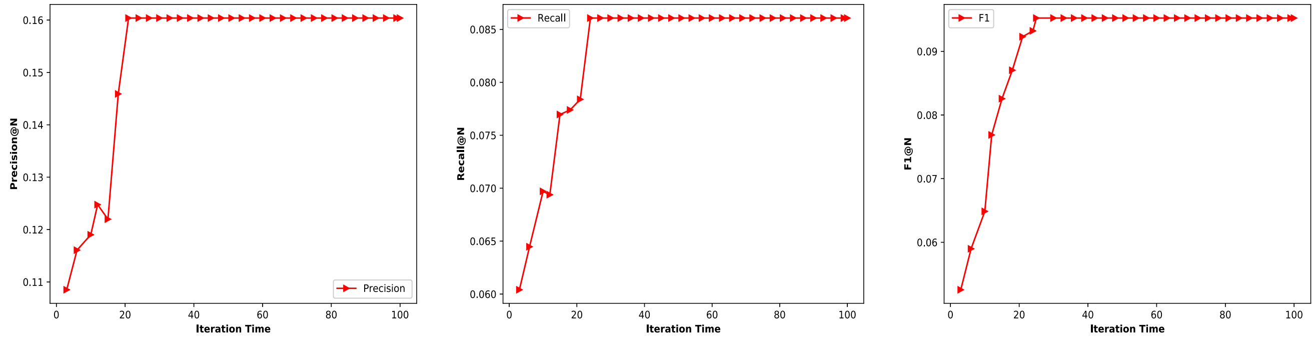
(a) Influence of researcher's degree on precision (b) Influence of researcher's degree on recall (c) Influence of researcher's degree on F1

Fig. 9. Influence of target researcher's degree on precision, recall and F1 (hep-th).



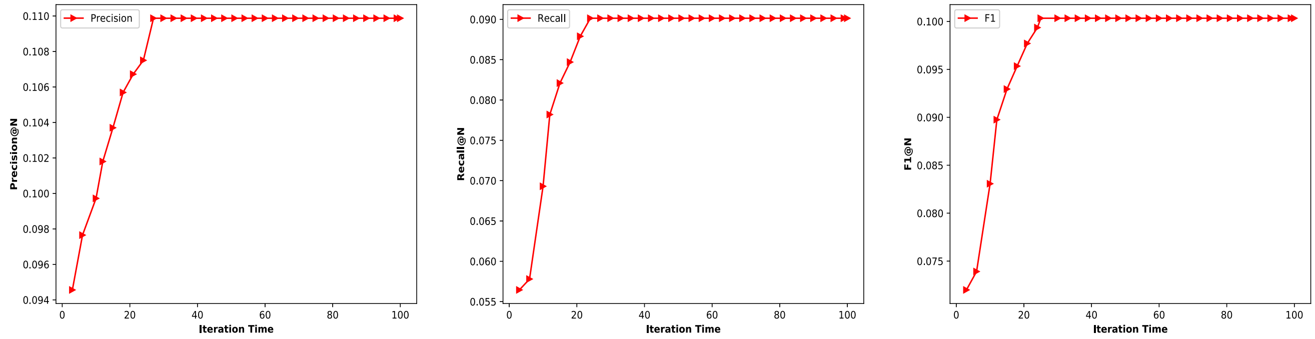
(a) Influence of researcher's degree on precision (b) Influence of researcher's degree on recall (c) Influence of researcher's degree on F1

Fig. 10. Influence of target researcher's degree on precision, recall and F1 (DBLP).



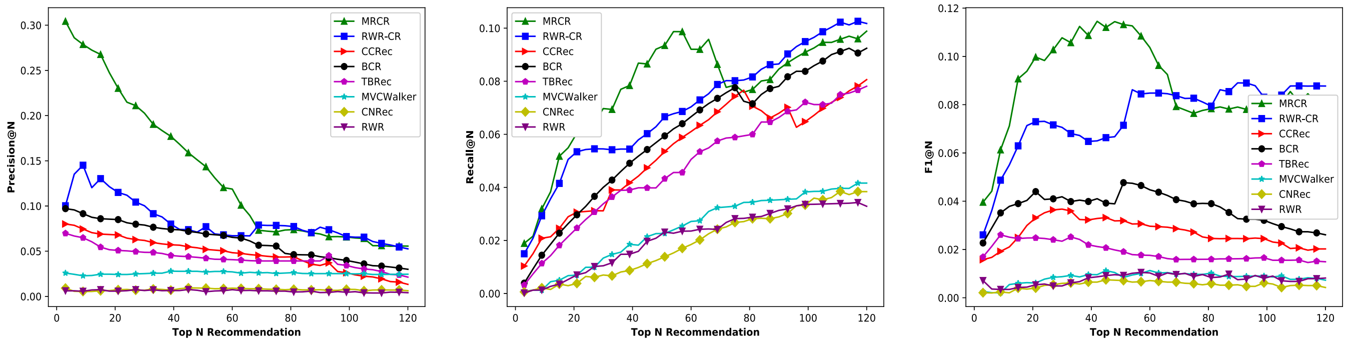
(a) Influence of iteration on precision (b) Influence of iteration on recall (c) Influence of iteration on F1

Fig. 11. Influence of iteration on precision, recall, and F1 (hep-th).



(a) Influence of iteration on precision (b) Influence of iteration on recall (c) Influence of iteration on F1

Fig. 12. Influence of iteration on precision, recall, and F1(DBLP).



(a) Precision of MRCR (b) Recall of MRCR (c) F1 of MRCR

Fig. 13. MRCR performance analysis in terms of precision, recall and F1 (hep-th).

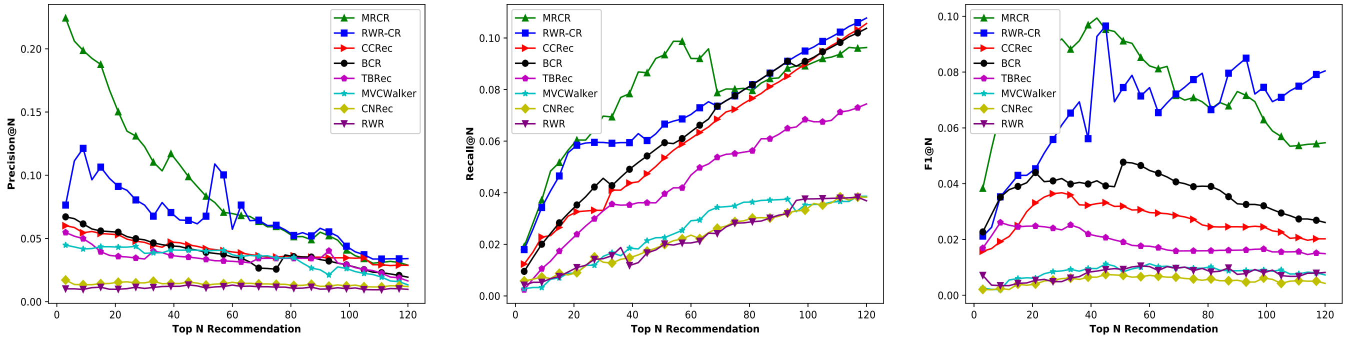
performs an upward trend from the beginning and achieves the highest F1 of 0.993 at position 42. It shows a downward trend and reaches a F1 of 0.069 at position 72. Then it shows a steady performance over the recommendation list and finally reaches a F1 of 0.054 at position 120 as shown in Fig. 14(c). During the initial recommendation, the MRCR model made a significant improvement on evaluation metrics, such as precision, recall and F1 over the standard approaches.

11.2. Results analysis of DBCR model

The detailed results are shown in Figs. 15 and 16 respectively. We have experimented on the hep-th dataset, and observed that DBCR model, achieves the highest precision of 0.098 at first recommendation and slowly shows a downward trend and reaches a precision value of 0.084 at position 120 as shown in Fig. 15(a). Similarly, it achieves the highest precision of 0.073 at first recommendation and shows a downward trend and reaches a value of 0.047 at position 120 on DBLP dataset as shown in Fig. 16(a).

In the case of recall evaluation on hep-th, the DBCR model performs an upward trend and reaches a recall of 0.101 at position 56 and then slowly increases and achieves a recall of 0.114 at position 120 as shown in Fig. 15(b). The DBCR model shows a similar nature of performance on DBLP dataset too. As shown in Fig. 16(b), it recommends with a higher recall of 0.099 at position 56, and afterward, it seems like a trend of decline on recall to a certain degree and reaches a recall of 0.105 while recommending 120 collaborators.

Similarly, while evaluating the performance on hep-th, the DBCR model performs an upward trend from the beginning and achieves the highest F1 of 0.111 at position 50. Then it shows

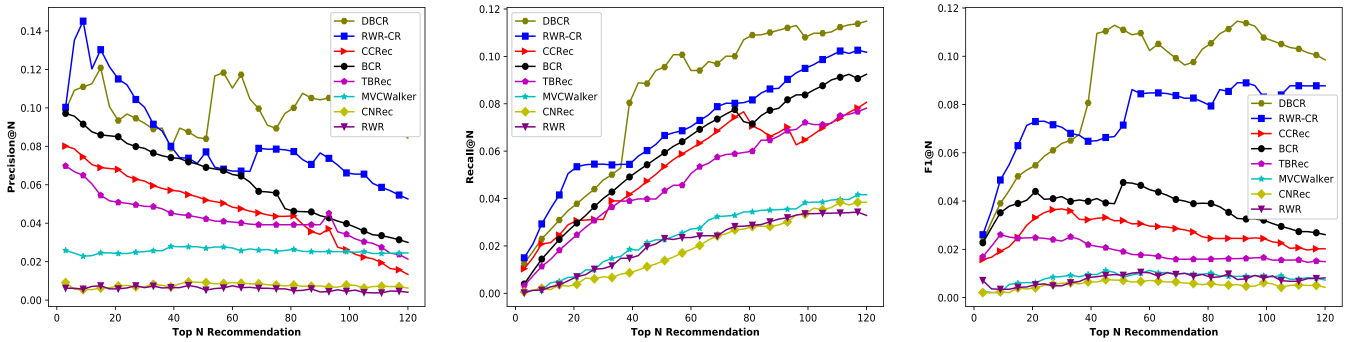


(a) Precision of MRCR

(b) Recall of MRCR

(c) F1 of MRCR

Fig. 14. MRCR performance analysis in terms of precision, recall and F1 (DBLP).

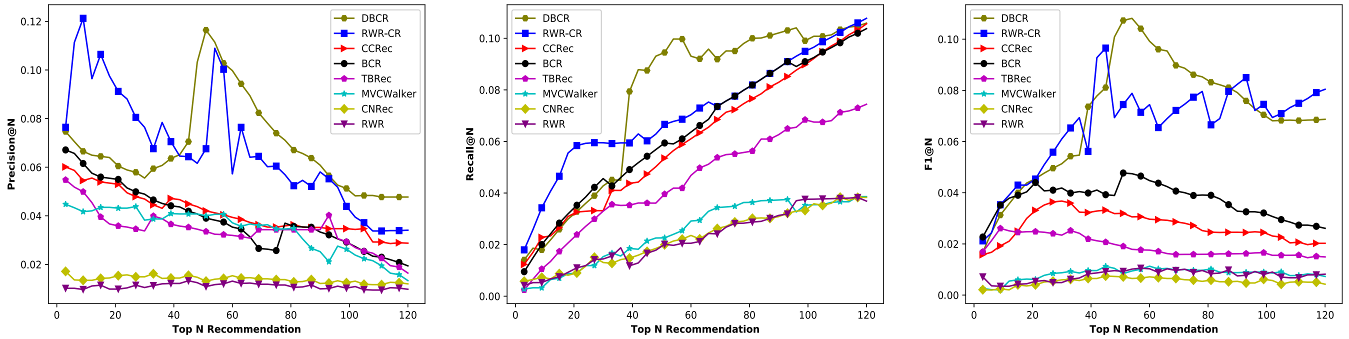


(a) Precision of DBCR

(b) Recall of DBCR

(c) F1 of DBCR

Fig. 15. DBCR performance analysis in terms of precision, recall and F1 (hep-th).



(a) Precision of DBCR

(b) Recall of DBCR

(c) F1 of DBCR

Fig. 16. DBCR performance analysis in terms of precision, recall and F1 (DBLP).

a downward trend and reaches a F1 of 0.098 at position 120 as shown in Fig. 15(c). But in case of DBLP dataset, the model shows an upward trend from the beginning and reaches a F1 of 0.106 at position 50, and it shows a downward trend slowly and finally achieves a F1 of 0.068 at position 120 as shown in Fig. 16(c).

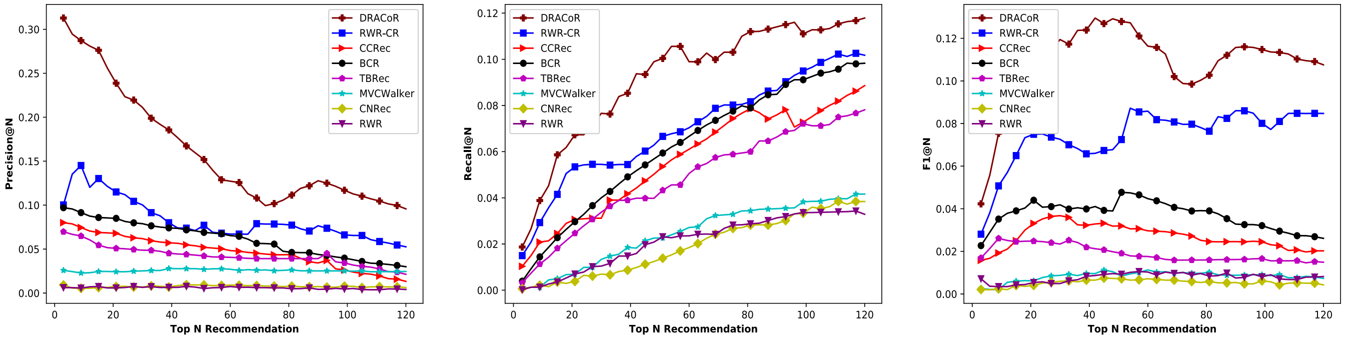
We observed that during the mid-end stages of the recommendation, DBCR model made a significant improvement on evaluation metrics like precision, recall and F1 over the standard approaches.

11.3. Results analysis of DRACoR model

The detailed results are shown in Figs. 17 and 18 respectively. We have experimented on hep-th dataset and observed that proposed model DRACoR exhibits the highest precision of 0.287 after recommending the top 10 potential collaborators and then slowly it shows a downward trend and reaches a precision value of 0.095 at position 120 as shown in Fig. 17(a). Similarly, it achieves the highest precision of 0.207 at position 10 and then slowly shows a downward trend and reaches a precision value of 0.051 at position 120 on DBLP dataset as shown in Fig. 18(a).

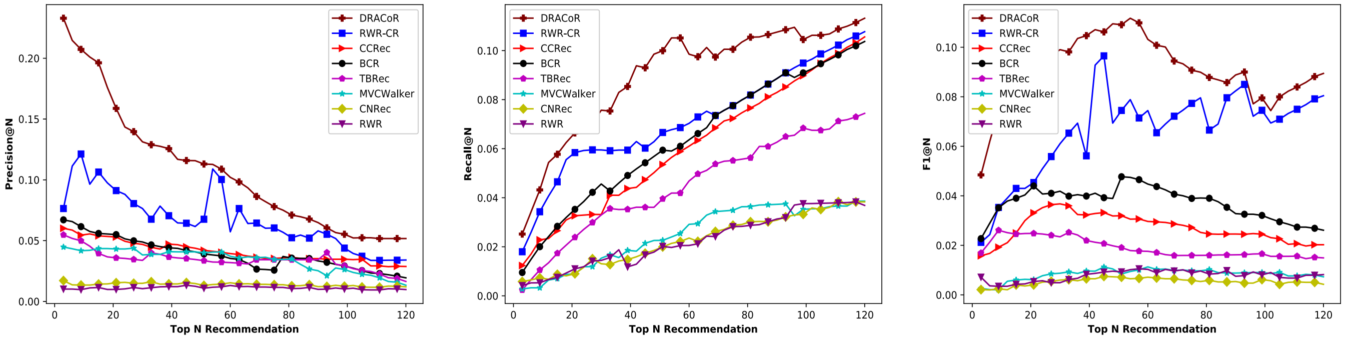
In case of recall evaluation on hep-th, the proposed model DRACoR performs an upward trend and reaches a recall of 0.0987 at position 54 and then slowly increases and achieves a recall of 0.117 at position 120 as shown in Fig. 17(b). The DRACoR model shows a similar nature of performance on DBLP dataset too. As shown in Fig. 18(b), it recommends with a higher recall of 0.098 at position 60 and finally reaches a recall of 0.113 while recommending 120 collaborators.

Similarly, while evaluating F1 on hep-th the DRACoR model performs an upward trend from the beginning and achieves a F1 of 0.127 at position 50. Then it shows a downward trend and



(a) Precision of DRACoR (b) Recall of DRACoR (c) F1 of DRACoR

Fig. 17. DRACoR performance analysis in terms of precision, recall and F1 (hep-th).



(a) Precision of DRACoR (b) Recall of DRACoR (c) F1 of DRACoR

Fig. 18. DRACoR performance analysis in terms of precision, recall and F1 (DBLP).

reaches a F1 of 0.107 at position 120 as shown in Fig. 17(c). But in case of DBLP dataset the model shows an upward trend from the beginning and reaches a F1 of 0.109 at position 50, and slowly it shows a downward trend and finally achieves a F1 of 0.089 at position 120 as shown in Fig. 18(c).

The complete results of MRR and nDCG are depicted in Tables 12 and 13. It is evident from Table 12 that, proposed model DRACoR shows a consistent nDCG and MRR over all other standard approaches on hep-th dataset. The proposed approach shows an MRR of 0.4578 indicates the effectiveness of correctly predicting the first collaborators within top 2 recommendations. While evaluating the performance analysis in terms of nDCG, DRACoR shows the highest nDCG of 0.2993 at position 10 then slowly it decreases and achieves an nDCG of 0.1509 at position 120 as shown in Table 12.

In the case of MRR evaluation on the DBLP dataset, DRACoR shows an MRR of 0.4109, which indicates the effectiveness of correctly predicting the first collaborators within the top 2 recommendations. Similarly, for nDCG evaluation on DBLP, it is visible that the proposed model DRACoR exhibits a significant improvement of nDCG over all other state-of-the-art methods. It is clearly shown in Table 13 that the nDCG results of DRACoR are consistent and show the highest nDCG of 0.2793 at position 10 and display the worst nDCG of 0.1406 at position 120.

We also conduct pairwise t-tests on overall F1, MRR, and nDCG for both hep-th and DBLP datasets between DRACoR and the best among state-of-the-art methods at 5% level of significance. The complete results are shown in Tables 10–13 respectively.

12. Study of the proposed approach

The main findings concerning our various RQs are summarized below.

12.1. RQ1: How does different parameter selection affect the performance of DRACoR?

We have evaluated the impact of various experimental parameter settings, including vector dimension (A_i and T_i), adjustment parameter (m), damping constant (α), target researcher's academic level (n_c), target researcher's degree (n_d), number of iteration, and partitioning time point on DRACoR. The overall results are shown in Section 10.5.

12.2. RQ2: How does DRACoR handle the cold-start issue for the new researcher?

We conducted an extensive experiment to prove the efficacy of the proposed model DRACoR against new collaborators. Our model recommends collaborators, including new and old irrespective of target researchers' degree and academic level. To validate the effectiveness of DRACoR, we experimented with varying academic level (n_c) of a target researcher as explained in Section 10.5.4. We also evaluated with varying target researcher's degree (n_d) as described in Section 10.5.5.

12.3. RQ3: How effective is DRACoR in comparison to other state-of-the-art methods?

The complete results of F1 are depicted in Tables 10 and 11. It is evident that the proposed approach DRACoR shows a consistent F1 over all other standard approaches on the hep-th dataset and DBLP dataset. The complete results of MRR and nDCG are depicted in Tables 12 and 13. It is evident that, proposed approach shows a consistent nDCG and MRR over all other standard approaches on hep-th dataset and DBLP dataset.

Table 10
F1-score results of MRCR, DBCR and other approaches (hep-th)

Methods	F1@10	F1@20	F1@30	F1@40	F1@50	F1@60	F1@80	F1@100	F1@120
CNRec	0.0024	0.0040	0.0059	0.0064	0.0071	0.0071	0.0058	0.0061	0.0042
RWR	0.0034	0.0053	0.0048	0.0082	0.0092	0.0102	0.0083	0.0080	0.0081
MVCWalker	0.0020	0.0061	0.0087	0.0096	0.0084	0.0112	0.0101	0.0090	0.0072
TBRec	0.0260	0.0248	0.0234	0.0219	0.0190	0.0175	0.0159	0.0165	0.0149
CCRec	0.0192	0.0331	0.0367	0.0322	0.0319	0.0296	0.0245	0.0244	0.0202
BCR	0.0351	0.0439	0.0418	0.0398	0.0477	0.0448	0.0390	0.0320	0.0260
RWR-CR	0.0487^{+}	0.0729^{+}	0.0706^{+}	0.0647^{+}	0.0714^{+}	0.0848^{+}	0.0793^{+}	0.0833^{+}	0.0877^{+}
MRCR	0.0612	0.0997	0.1077	0.1086	0.1131	0.1037	0.0783	0.0795	0.0793
DBCR	0.0390	0.0548	0.0639	0.0806	0.1109	0.1023	0.1053	0.1077	0.0984
DRACoR	0.0753*	0.1120*	0.1194*	0.1238*	0.1278*	0.1163*	0.1026*	0.1147*	0.1075*

*Denotes statistical significance ($\alpha = 0.05$) over the best among state-of-the-art ('+')

Table 11
F1-score results of MRCR, DBCR and other approaches (DBLP)

Methods	F1@10	F1@20	F1@30	F1@40	F1@50	F1@60	F1@80	F1@100	F1@120
CNRec	0.0021	0.0038	0.0057	0.0061	0.0074	0.0066	0.0061	0.0062	0.0040
RWR	0.0032	0.0051	0.0045	0.0079	0.0089	0.0100	0.0081	0.0077	0.0076
MVCWalker	0.0018	0.0058	0.0084	0.0093	0.0082	0.0109	0.0098	0.0087	0.0069
TBRec	0.0257	0.0246	0.0231	0.0215	0.0186	0.0172	0.0157	0.0161	0.0145
CCRec	0.0188	0.0326	0.0363	0.0318	0.0314	0.0291	0.0242	0.0239	0.0201
BCR	0.0348	0.0433	0.0413	0.0393	0.0472	0.0443	0.0384	0.0315	0.0256
RWR-CR	0.0352^{+}	0.0453^{+}	0.0610^{+}	0.0561^{+}	0.0744^{+}	0.0744^{+}	0.0665^{+}	0.0745^{+}	0.0804^{+}
MRCR	0.0656	0.0891	0.0918	0.0969	0.0911	0.0822	0.0664	0.0629	0.0546
DBCR	0.0312	0.0455	0.0513	0.0736	0.1073	0.0991	0.0831	0.0704	0.0686
DRACoR	0.0744*	0.0966*	0.0990*	0.1046*	0.1090*	0.1031*	0.0878*	0.0795*	0.0894*

*Denotes statistical significance ($\alpha = 0.05$) over the best among state-of-the-art ('+').

Table 12
MRR and nDCG results of DRACoR and other approaches (hep-th)

Methods	MRR	nDCG@10	nDCG@20	nDCG@30	nDCG@40	nDCG@60	nDCG@80	nDCG@100	nDCG@120
CNRec	0.1615	0.0055	0.0054	0.0049	0.0056	0.0049	0.0045	0.0041	0.0039
RWR	0.1798	0.0228	0.0219	0.0174	0.0156	0.0127	0.0119	0.0096	0.0082
MVCWalker	0.1974	0.0239	0.0248	0.0277	0.0269	0.0268	0.0258	0.0226	0.0197
TBRec	0.2208	0.0508	0.0487	0.0452	0.0406	0.0392	0.0332	0.0249	0.0231
CCRec	0.0679	0.0673	0.0548	0.0477	0.0429	0.0358	0.0341	0.0324	0.0319
BCR	0.1971	0.0775	0.0746	0.0668	0.0644	0.0697^{+}	0.0473	0.0597^{+}	0.0496^{+}
RWR-CR	0.2247^{+}	0.1703^{+}	0.1687^{+}	0.1459^{+}	0.1002^{+}	0.0695	0.0639^{+}	0.0591	0.0492
MRCR	0.4292	0.2576	0.2663	0.2669	0.2108	0.1749	0.1386	0.1252	0.1196
DBCR	0.2038	0.1645	0.1856	0.1747	0.1793	0.1966	0.1686	0.1593	0.1478
DRACoR	0.4578*	0.2993*	0.2892*	0.2886*	0.2372*	0.2019*	0.1837*	0.1693*	0.1509*

*Denotes statistical significance ($\alpha = 0.05$) over the best among state-of-the-art ('+')

Table 13
MRR and nDCG results of DRACoR and other approaches (DBLP)

Methods	MRR	nDCG@10	nDCG@20	nDCG@30	nDCG@40	nDCG@60	nDCG@80	nDCG@100	nDCG@120
CNRec	0.1536	0.0042	0.0049	0.0052	0.0055	0.0047	0.0041	0.0039	0.0035
RWR	0.1589	0.0215	0.0219	0.0169	0.0149	0.0121	0.0115	0.0089	0.0079
MVCWalker	0.1878	0.0219	0.0235	0.0265	0.0251	0.0255	0.0247	0.0219	0.0189
TBRec	0.2338^{+}	0.0497	0.0469	0.0431	0.0389	0.0392	0.0319	0.0225	0.0218
CCRec	0.0643	0.0652	0.0519	0.0439	0.0407	0.0348	0.0331	0.0309	0.0298
BCR	0.1877	0.0691	0.0729	0.0615	0.0598	0.0654	0.0435	0.0516	0.0448
RWR-CR	0.1975	0.1694^{+}	0.1589^{+}	0.1424^{+}	0.0984^{+}	0.0583^{+}	0.0616^{+}	0.0536^{+}	0.0467^{+}
MRCR	0.4005	0.2449	0.2573	0.2557	0.2012	0.1674	0.1211	0.1226	0.1098
DBCR	0.1985	0.1544	0.1769	0.1693	0.1671	0.1849	0.1537	0.1493	0.1368
DRACoR	0.4109*	0.2793*	0.2787*	0.2804*	0.2295*	0.1982*	0.1768*	0.1578*	0.1406*

*Denotes statistical significance ($\alpha = 0.05$) over the best among state-of-the-art ('+').

12.4. More insightful discussion on the results

The overall performance results obtained showcase the efficacy of the proposed DRACoR. The good overall precision, recall, F1, MRR, and nDCG verify that the proposed model DRACoR can effectively recommend the relevant collaborators. However, there are a few limitations to our work.

(i) As we have considered only top 100 topics for each researcher. As a result, it may fail to recommend relevant collaborators where a target researcher is associated with multiple research areas.

(ii) We do not consider the affiliation data, due to which in few cases both MRCR and DBCR models exhibit the worst performance in a few positions over state-of-the-art methods.

(iii) We have considered multiple factors to enhance the link importance among researchers but the individual MRCR or DBCR model is not stable throughout the recommendation.

Please cite this article as: T. Pradhan and S. Pal, A multi-level fusion based decision support system for academic collaborator recommendation, Knowledge-Based Systems (2020) 105784, https://doi.org/10.1016/j.knosys.2020.105784.

We also observed that the MRCR model shows better performance until position 60. But afterward, it exhibits the worst performance over other standard methods.

(iv) As we adopted RWR model to recommend collaborators in MRCR Model which can jump with a probability of α, and restart probability of 1-α. We have set the value of α as 0.8 due to which after recommending 60 collaborators, the chances of getting relevant researchers are quite rare. The chances of getting other researchers (researcher with other research areas) will be more, and this might be the reason for obtaining the worst results after position 60 in MRCR model.

(v) Although we have used deep learning in DBCR model to capture hidden relationships mostly, the model performs worst up to position 30, and afterward, it displays effective results over state-of-the-art methods. Mainly deep learning is biased to content similarity and due to which it recommends collaborators with a similar area of research. Besides, there can be many latent reasons for future collaboration other than topic similarity.

13. Conclusion

In this paper, we focus on recommending MICs (MPCs+MVCs), which can help researchers benefit more from collaboration based on the big scholarly data. We mainly focused on recommending potential collaborators based on similar research interests and social accessibility. We propose a multi-level fusion-based academic collaborator recommender system DRACoR. Mainly, it fuses Meta-path aggregated Random walk based Collaborator Recommendation (MRCR) that finds out MPCs with Deep learning-Boosted Collaborator Recommendation (DBCR) models that find MVCs so that their combination (MICs) can be recommended. The proposed model DRACoR works irrespective of researchers' past publication records and is entirely biased towards the current works. Isolated researchers, researchers with less number of co-authors, or researchers with fewer publication records are also getting an equal chance of inclusion in the final recommendation.

Individually, we have considered a few factors, namely meta-path features, dynamic interest, research content, scholarly influence-aware features, and hidden relationship to determine the similarity between two researchers. We explored two influence-aware features such as H-index based level similarity and percentage of collaboration to enhance the link importance among researchers.

We conducted extensive experiments on a subset of hep-th and DBLP datasets to evaluate the performance of DRACoR against various state-of-the-art methods. The proposed system DRACoR outperforms other state-of-the-art models when compared in terms of precision, recall, F1, MRR, and nDCG, respectively. The proposed model reveals that the combination of topic distribution and co-authorship networks based models can significantly improve the effectiveness of the academic collaborations.

Nonetheless, there is still room for future studies in this direction. Besides, there can be many latent reasons behind the collaboration of two researchers. They might have met at a meeting or are from the same institution. Additionally, many other features such as researcher age, education, institution, acknowledgment details, the personal profile should be explored to improve upon our model. Collaboration can also be for cross-domain research. The relationship among co-authors of a paper is far more complicated than what we have imagined. As for future work, more experiments on the other large bibliographic datasets could be conducted to validate the effectiveness and practicability.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] J. Lu, D. Wu, M. Mao, W. Wang, G. Zhang, Recommender system application developments: a survey, Decis. Support Syst. 74 (2015) 12–32.

[2] D. Liang, L. Charlin, J. McInerney, D.M. Blei, Modeling user exposure in recommendation, in: Proceedings of the 25th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2016, pp. 951–961.

[3] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, IEEE Trans. Knowl. Data Eng. 17 (6) (2005) 734–749.

[4] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender systems survey, Knowl. Based Syst. 46 (2013) 109–132.

[5] J. Tang, S. Wu, J. Sun, H. Su, Cross-domain collaboration recommendation, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2012, pp. 1285–1293.

[6] S. Cohen, L. Ebel, Recommending collaborators using keywords, in: Proceedings of the 22nd International Conference on World Wide Web, ACM, 2013, pp. 959–962.

[7] P. Chaiwanarom, C. Lursinsap, Collaborator recommendation in interdisciplinary computer science using degrees of collaborative forces, temporal evolution of research interest, and comparative seniority status, Knowl.-Based Syst. 75 (2015) 161–172.

[8] J. Son, S.B. Kim, Academic paper recommender system using multilevel simultaneous citation networks, Decis. Support Syst. 105 (2018) 24–33.

[9] Y. Sebastian, E.-G. Siew, S.O. Orimaye, Learning the heterogeneous bibliographic information network for literature-based discovery, Knowl.-Based Syst. 115 (2017) 66–79.

[10] G. Wang, X. He, C.I. Ishuga, HAR-SI: A novel hybrid article recommendation approach integrating with social information in scientific social network, Knowl.-Based Syst. 148 (2018) 85–99.

[11] A.S. Raamkumar, S. Foo, N. Pang, Using author-specified keywords in building an initial reading list of research papers in scientific paper retrieval and recommender systems, Inf. Process. Manage. 53 (3) (2017) 577–594.

[12] W. Huang, Z. Wu, L. Chen, P. Mitra, C.L. Giles, A neural probabilistic model for context based citation recommendation, in: AAAI, 2015, pp. 2404–2410.

[13] X. Liu, Y. Yu, C. Guo, Y. Sun, Meta-path-based ranking with pseudo relevance feedback on heterogeneous graph for citation recommendation, in: Proceedings of the 23rd Acm International Conference on Conference on Information and Knowledge Management, ACM, 2014, pp. 121–130.

[14] X. Liu, J. Zhang, C. Guo, Citation recommendation via proximity full-text citation analysis and supervised topical prior, IConference 2016 Proceedings, iSchools, 2016.

[15] Q. He, D. Kifer, J. Pei, P. Mitra, C.L. Giles, Citation recommendation without author supervision, in: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, ACM, 2011, pp. 755–764.

[16] J. Beel, B. Gipp, S. Langer, C. Breitinger, Paper recommender systems: a literature survey, Int. J. Digital Libraries 17 (4) (2016) 305–338.

[17] Z. Yang, D. Yin, B.D. Davison, Recommendation in academia: A joint multi-relational model, in: Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on, IEEE, 2014, pp. 566–571.

[18] F. Xia, N.Y. Asabere, J.J. Rodrigues, F. Basso, N. Deonauth, W. Wang, Socially-aware venue recommendation for conference participants, in: Ubiquitous Intelligence and Computing, 2013 IEEE 10th International Conference on and 10th International Conference on Autonomic and Trusted Computing, UIC/ATC, IEEE, 2013, pp. 134–141.

[19] F. Xia, W. Wang, T.M. Bekele, H. Liu, Big scholarly data: A survey, IEEE Trans. Big Data 3 (1) (2017) 18–35.

[20] S. Yu, J. Liu, Z. Yang, Z. Chen, H. Jiang, A. Tolba, F. Xia, PAVE: Personalized academic venue recommendation exploiting co-publication networks, J. Netw. Comput. Appl. 104 (2018) 38–47.

[21] J. Katz, B.R. Martin, What is research collaboration? Res. policy 26 (1) (1997) 1–18.

[22] S. Lee, B. Bozeman, The impact of research collaboration on scientific productivity, Soc. Stud. Sci. 35 (5) (2005) 673–702.

[23] G. George, M.R. Haas, A. Pentland, Big Data and Management, Academy of Management Briarcliff Manor, NY, 2014.

[24] M. Mirzaei, J. Sander, E. Stroulia, Multi-aspect review-team assignment using latent research areas, Inf. Process. Manage. 56 (3) (2019) 858–878.

[25] C. Xu, A novel recommendation method based on social network using matrix factorization technique, Inf. Process. Manage. 54 (3) (2018) 463–474.

[26] D. Wang, Y. Liang, D. Xu, X. Feng, R. Guan, A content-based recommender system for computer science publications, Knowl.-Based Syst. 157 (2018) 1–9.

[27] H.P. Luong, T. Huynh, S. Gauch, K. Hoang, Exploiting social networks for publication venue recommendations, in: KDIR, 2012, pp. 239–245.

[28] X. Zhou, L. Ding, Z. Li, R. Wan, Collaborator recommendation in heterogeneous bibliographic networks using random walks, Inform. Retriev. J. 20 (4) (2017) 317–337.

[29] X. Kong, H. Jiang, W. Wang, T.M. Bekele, Z. Xu, M. Wang, Exploring dynamic research interest and academic influence for scientific collaborator recommendation, Scientometrics 113 (1) (2017) 369–385.

[30] G.R. Lopes, M.M. Moro, L.K. Wives, J.P.M. De Oliveira, Collaboration recommendation on academic social networks, in: International Conference on Conceptual Modeling, Springer, 2010, pp. 190–199.

[31] D.H. Lee, P. Brusilovsky, T. Schleyer, Recommending collaborators using social features and mesh terms, Proc. Amer. Soc. Inform. Sci. Technol. 48 (1) (2011) 1–10.

[32] I. Konstas, V. Stathopoulos, J.M. Jose, On social networks and collaborative recommendation, in: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2009, pp. 195–202.

[33] X. Kong, H. Jiang, Z. Yang, Z. Xu, F. Xia, A. Tolba, Exploiting publication contents and collaboration networks for collaborator recommendation, PLoS One 11 (2) (2016) e0148492.

[34] J. Li, F. Xia, W. Wang, Z. Chen, N.Y. Asabere, H. Jiang, Acrec: a co-authorship based random walk model for academic collaboration recommendation, in: Proceedings of the 23rd International Conference on World Wide Web, ACM, 2014, pp. 1209–1214.

[35] F. Xia, Z. Chen, W. Wang, J. Li, L.T. Yang, Mvcwalker: Random walk-based most valuable collaborators recommendation exploiting academic factors, IEEE Trans. Emerg. Top. Comput. 2 (3) (2014) 364–375.

[36] L. Deng, D. Yu, et al., Deep learning: methods and applications, Found. Trends® Signal Process. 7 (3–4) (2014) 197–387.

[37] G.R. Lopes, M.M. Moro, L.K. Wives, J.P.M. de Oliveira, Collaboration recommendation on academic social networks, in: J. Trujillo, G. Dobbie, H. Kangassalo, S. Hartmann, M. Kirchberg, M. Rossi, I. Reinhartz-Berger, E. Zimányi, F. Frasincar (Eds.), Advances in Conceptual Modeling – Applications and Challenges, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 190–199.

[38] S.D. Gollapalli, P. Mitra, C. Giles, Similar researcher search in academic environments, in: Proceedings of the 12th ACM/IEEE-CS Joint Conference on Digital Libraries, ACM, 2012, pp. 167–170.

[39] C. Yang, J. Ma, J. Sun, T. Silva, X. Liu, Z. Hua, A weighted topic model enhanced approach for complementary collaborator recommendation, in: PACIS, 2014, p. 297.

[40] Z. Liu, X. Xie, L. Chen, Context-aware academic collaborator recommendation, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM, 2018, pp. 1870–1879.

[41] W. Glänzel, A. Schubert, Analysing scientific networks through co-authorship, in: Handbook of Quantitative Science and Technology Research, Springer, 2004, pp. 257–276.

[42] M.E. Newman, Scientific collaboration networks. I. Network construction and fundamental results, Physical review E 64 (1) (2001) 016131.

[43] A.-L. Barabâsi, H. Jeong, Z. Néda, E. Ravasz, A. Schubert, T. Vicsek, Evolution of the social network of scientific collaborations, Physica A: Statistical mechanics and its applications 311 (3–4) (2002) 590–614.

[44] X. Liu, J. Bollen, M.L. Nelson, H. Van de Sompel, Co-authorship networks in the digital library research community, Inform. Process. Manag. 41 (6) (2005) 1462–1480.

[45] M. Jamali, M. Ester, Trustwalker: a random walk model for combining trust-based and item-based recommendation, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2009, pp. 397–406.

[46] L. Backstrom, J. Leskovec, Supervised random walks: predicting and recommending links in social networks, in: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, ACM, 2011, pp. 635–644.

[47] X. Zhou, W. Liang, I. Kevin, K. Wang, R. Huang, Q. Jin, Academic influence aware and multidimensional network analysis for research collaboration navigation based on scholarly big data, IEEE Trans. Emerg. Top. Comput. (2018).

[48] W. Wang, J. Liu, Z. Yang, X. Kong, F. Xia, Sustainable collaborator recommendation based on conference closure, IEEE Trans. Comput. Soc. Syst. 6 (2) (2019) 311–322.

[49] N. Sun, Y. Lu, Y. Cao, Career age-aware scientific collaborator recommendation in scholarly big data, IEEE Access 7 (2019) 136036–136045.

[50] H.-H. Chen, L. Gou, X. Zhang, C.L. Giles, Collabseer: a search engine for collaboration discovery, in: Proceedings of the 11th Annual International ACM/IEEE Joint Conference on Digital Libraries, ACM, 2011, pp. 231–240.

[51] X. Kong, M. Mao, J. Liu, B. Xu, R. Huang, Q. Jin, Tnerec: Topic-aware network embedding for scientific collaborator recommendation, in: 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation, SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI, IEEE, 2018, pp. 1007–1014.

[52] C. Yang, J. Sun, J. Ma, S. Zhang, G. Wang, Z. Hua, Scientific collaborator recommendation in heterogeneous bibliographic networks, in: 2015 48th Hawaii International Conference on System Sciences, IEEE, 2015, pp. 552–561.

[53] Y. Sun, B. Norick, J. Han, X. Yan, P.S. Yu, X. Yu, Pathselclus: Integrating meta-path selection with user-guided object clustering in heterogeneous information networks, ACM Trans. Knowl. Discov. Data (TKDD) 7 (3) (2013) 11.

[54] Y. Sun, J. Han, Mining heterogeneous information networks: a structural analysis approach, ACM SIGKDD Explor. Newslett. 14 (2) (2013) 20–28.

[55] C. Shi, B. Hu, W.X. Zhao, S. Philip, Heterogeneous information network embedding for recommendation, IEEE Trans. Knowl. Data Eng. 31 (2) (2018) 357–370.

[56] Y. Sun, J. Han, Mining heterogeneous information networks: principles and methodologies, Synth. Lect. Data Mining Knowl. Discov. 3 (2) (2012) 1–159.

[57] Y. Sun, R. Barber, M. Gupta, C.C. Aggarwal, J. Han, Co-author relationship prediction in heterogeneous bibliographic networks, in: Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on, IEEE, 2011, pp. 121–128.

[58] A. Grover, J. Leskovec, node2vec: Scalable Feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 855–864.

[59] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (Jan) (2003) 993–1022.

[60] J.H. Lau, T. Baldwin, An empirical evaluation of doc2vec with practical insights into document embedding generation, 2016, ArXiv preprint, arXiv:1607.05368.

[61] Q. Le, T. Mikolov, Distributed representations of sentences and documents, in: International Conference on Machine Learning, 2014, pp. 1188–1196.

[62] Q.V. Le, T. Mikolov, Distributed representations of sentences and documents, CoRR abs/1405.4053 (2014) URL arXiv:1405.4053.

[63] Y. Sun, J. Han, X. Yan, P.S. Yu, T. Wu, Pathsim: Meta path-based top-k similarity search in heterogeneous information networks, Proc. VLDB Endow. 4 (11) (2011) 992–1003.

[64] I.A. Basheer, M. Hajmeer, Artificial neural networks: fundamentals, computing, design, and application, J. Microbiol. Methods 43 (1) (2000) 3–31.

[65] M.F. Porter, Snowball: A Language for Stemming Algorithms, 2001.

[66] R. Real, J.M. Vargas, The probabilistic basis of Jaccard's index of similarity, System. Biol. 45 (3) (1996) 380–385.

[67] H. Tong, C. Faloutsos, J.-Y. Pan, Fast Random Walk with Restart and Its Applications, IEEE, 2006.

[68] M. Van Setten, Supporting People in Finding Information: Hybrid Recommender Systems and Goal-based Structuring, 2005.

[69] S.-Y. Hwang, C.-P. Wei, Y.-F. Liao, Coauthorship networks and academic literature recommendation, Electron. Commer. Res. Appl. 9 (4) (2010) 323–334.

[70] X. Liu, X. Zhu, M. Li, L. Wang, E. Zhu, T. Liu, M. Kloft, D. Shen, J. Yin, W. Gao, Multiple kernel k-means with incomplete kernels, IEEE Trans. Pattern Anal. Mach. Intell (2019).

[71] J. Sun, J. Ma, Z. Liu, Y. Miao, Leveraging content and connections for scientific article recommendation in social computing contexts, Comput. J. 57 (9) (2014) 1331–1342.

[72] S. Wu, Applying the data fusion technique to blog opinion retrieval, Expert Syst. Appl. 39 (1) (2012) 1346–1353.

[73] D. Lillis, L. Zhang, F. Toolan, R.W. Collier, D. Leonard, J. Dunnion, Estimating probabilities for effective data fusion, in: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2010, pp. 347–354.

[74] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, Z. Su, Arnetminer: extraction and mining of academic social networks, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2008, pp. 990–998.

[75] H. Stuckenschmidt, Approximate information filtering on the semantic web, in: Annual Conference on Artificial Intelligence, Springer, 2002, pp. 114–128.

Please cite this article as: T. Pradhan and S. Pal, A multi-level fusion based decision support system for academic collaborator recommendation, Knowledge-Based Systems (2020) 105784, https://doi.org/10.1016/j.knosys.2020.105784.