Multilevel Graph-Based Decision Making in Big Scholarly Data: An Approach to Identify Expert Reviewer, Finding Quality Impact Factor, Ranking Journals and Researchers

M. Mazhar Rathore*, *Student Member, IEEE*, M. Junaid Gul, Anand Paul*, *Senior Member, IEEE,* Ashraf Ali Khan, Raja Wasim Ahmad, Joel J. P. C. Rodrigues, *Senior Member, IEEE*, Spiridon Bakiras

Abstract—Digital libraries, such as conference events, journal documents, books and thesis, research patents, and experiments generate a vast amount of data, named as, Scholarly Big Data. It covers scholarly related information for both researcher's perspective as well as publisher's perspective, such as academic activities, author's demography, academic social networks, etc. The relationships among Big Scholarly Data can be worthy of solving researcher as well as journal related concerns, if they are prudently treated to extract knowledge. The best approach to efficiently process these relationships is the graph. However, with the rapid growth in the number of digital articles by various libraries, the relationships raises exponentially, generating large graphs, which have become increasingly challenging to be handled in order to analyze scholarly information. On the other hand, many researchers and publishers/journals have severe concerns about the ranking control mechanisms and the consideration of quantity rather than quality. Therefore, in this paper, we proposed graph-based mechanisms to perform four critical decisions that are the need of the today's scholarly community. To improve the quality of the article, we proposed a mechanism for selecting and recommending suitable reviewers for a submitted paper based on researchers' expertise and their popularity in that particular field while avoiding conflict of interest. Also, due to shortcomings in the existing journal ranking approaches, we also designed a journal ranking mechanism including its new impact factor and relative ranking by using a modified version of traditional page ranking algorithm and excluding self-authors citations as well as self-journal citations. Similarly, researchers ranking is also important for various motives that is calculated based on the expert's field, citation count, and a number of publications while avoiding any loophole to increase the ranking such as, self-citations and wrong citations. Also, to efficiently process big graphs generated by a massive number of scholarly related relationships, we proposed an architecture that uses the parallel processing mechanism of the Hadoop ecosystem over the real-time analysis approach of Apache Spark with GraphX. Finally, the efficiency of the proposed system is evaluated in terms of processing time and throughput while implementing the designed decision mechanisms.

Index Terms—Big Scholarly Data, Big Graph, Hadoop, Apache Spark, Impact Factor, Journal and Conference Ranking.

1. Introduction

Science advances dramatically during the century. Scientists around the world are in a search of finding scientific answers to all the problems. Their quest primarily ends in the form of research articles. These facts and findings are shared with the world. The internet provides a good platform to share technical details about the research and the results. Many societies, journal and other venues are ready to publish research articles. With the addition to this, researchers have their own web pages to share briefed technical detail about their research. Thus, the volume of this research data is increasing day by day. Because of ever increasing volume of research articles, a new term scholarly Big Data (BSD) is now evolving. The 3Vs of big data (Volume, velocity & Variety) make this point stronger than ever that scholarly data is now big scholarly data. According to an article published in 2014 [1], 114 million research articles are now on the internet and adding up more at the speed of tens of thousands every day that confirms high volume and velocity of scholarly data[2]. Third "V" of big data is the variety of relationships among scholarly data, making it harder to analyze.

The issues related to big data, i.e. data management and analysis are also associated with BSD. Searching any document in the flood of data is not easy. This makes searching required information a hectic job. This situation arises the need of a new analysis system that can perform scientific methods and algorithms on BSD. In current scenario, the content, social, and statistical analysis are being used to analyze big scholarly data. Information like author relationships, common citations, finding domain expert etc., can be found by analyzing BSD. The reputations of the domain expert can also be analyzed by keeping some metrics and analyzing them accordingly. This information is not only limited to finding highest cited papers or authors but can also be used for future planning like resource allocating, finding suitable platform for publishing articles, and finding good sources for research.

Analyzing BSD provides variety of information about scholars, journals, and their interaction to one-another and to other institutions. However, still a very limited research work has been done in this area. The main factor for such ignorance might be the unavailability of tools and techniques which can

- *M. Mazhar Rathore is with the School of Computer Science and Engineering, Kyungpook National University, 80-Daehakro, Daegu, South Korea. E-mail: rathoremazhar@gmail.com*
- *M. Junaid Gul is with the School of Computer Science and Engineering, Kyungpook National University, 80-Daehakro, Daegu, South Korea. Email: junaidgul@live.com.pk*
- *Anand Paul is with the School of Computer Science and Engineering, Kyungpook National University 80-Daehakro, Daegu, South Korea. Email: Paul.editor@gmail.com*
- *Ashraf Ali Khan is with the The University of British Columbia, Canada. Email: khan.ashraf@ubc.ca*
- *Raja Wasim Ahmad is with the COMSATS Institute of IT, Pakistan and C4MCCR Malaysia. Email: wasimraja@ciit.net.pk*
- *Joel J. P. C. Rodrigues is with the National Institute of Telecommunications (Inatel), Santa Rita do Sapuca´ı-MG, Brazil, Instituto de Telecomunicac¸ ˜oes, Portugal, ITMO University, St. Petersburg, Russia, and University of Fortaleza (UNIFOR), Fortaleza-CE, Brazil. Email: joeljr@ieee.org*
- *Spiridon Bakiras is with the Division of Information and Computing Technology, College of Science and Engineering, Hamad Bin Khalifa University, Doha, Qatar. Email: sbakiras@hbku.edu.qa*

efficiently analyze big scholarly data. But, as new tools and techniques are now introducing, research can focus on BSD and can produce better results to understand facts about scholars, journals, and other research institutions. One of the best ways could be representing BSD in the graph. With the help of the graph, it might be easy to understand the relationship between certain entities. However, generating a graph for big data is itself not an easy task. Special tools and techniques are required.

A tool like Hadoop provides a platform that makes big data analysis bit easy. Information about impact factor, citation and finding domain expert can be derived after analyzing BSD with tools like Hadoop. Conversely, there are certain limitation in such scholarly measurements that make them unreliable. The main metric that is normally used to find reputation of a journal or an author is the journal Impact factor. The impact factor is derived from the statistical analysis of number citation of the research article and their impact on the research community [3]. Such metrics are commonly used [4] and few of them are more recent [5]. AS most of these metrics heavily rely on statistical analysis of the number of citations while neglecting the fact that the author can self-cite to gain more reputation. This scenario is now under the consideration by the research community to find a way to solve such problem scientifically.

Also, finding the domain expert to review the research article is also under debate. There are many scientific and academic procedures involved while publishing the research article [6]. Mainly, the paper is forwarded to the reviewer to give comments on its quality. Main problem in this process if to find the domain expert who can provide critical reviews according to the COPE Ethical Guidelines for Peer reviewer [7]. In addition to this American Geophysical Union (AGU) also include a summary of Scientific Integrity and Professional Ethics (SIPE) document on their website [7]. Many such ethical issues are also described by the Wijnholds and his companions [8]. Others related to user experience (QoE) and delay in media cloud is exploited by Liang Zhou [9, 10]. One of the critical points in research article publication is to find the appropriate reviewers which is often neglected by the editors, as sometimes it is hard to find domain expert from scholarly data.

Therefore, In order to cater the above-mentioned challenges faced by the research community, in this paper, we are proposing graph-based mechanisms to find out the answers to four critical questions that are raised by all research entities. We proposed a procedure to select and recommend suitable reviewers for a submitted research article based on researchers ' expertise and their reputation in particular domain while avoiding conflict of interest. We also propose a mechanism to evaluate new impact factor and relative ranking by using a modified version of traditional page ranking algorithm while eliminating self-author citations along with the self-journal citations. Assigning ranks and finding researchers' reputation is also important to various motives. These are calculated on the basis of expert's domain, number of citations, and a number of publications while eliminating any loophole that can affect the ranking process such as, self-citations and wrong citations. As big graphs are complicated to process, we are proposing to analyze these big graphs generated by a huge number of scholarly related relationships with parallel processing mechanism of Hadoop ecosystem along with the real-time analysis approach from Apache Spark with GraphX. To evaluate the efficiency of the proposed system, we considered processing time and throughput.

The Rest of the paper described the whole mechanism in detail. The paper is organized as follows. Section 2 presents the background knowledge and exiting work done in the related field. Section 3 shows how the big scholarly data is represented in graphs to get valuable knowledge efficiently. Section 4 described the proposed architecture that have the ability to process huge size of scholarly data in the form of graphs. Whereas, section 5 proposed various algorithms to find out the new impact factor, journal's ranking and researcher's ranking, expert reviewer selection, and conflict of interest. Finally, the proposed approach is implemented and evaluated, which is described in section 6. Section 7 concluded the article.

2. Background and Related Work

The concept of graph theory can be exploited to analyze big scholarly data. Graph's nodes and edges can represent relations (papers, with journals, with authors) among the scholars and journals. By analyzing BSD, finding domain expert should be easy and can ensure that research article sent to an expert that is somehow related to the authors. As an example, if we want to find a relation between author and coauthors, the graph can represent such relationships. Same scenario works in social networks and transportation [11, 12, 13, 14]. Like, if we want to find who like your Facebook account or post, the graph can reveal relationship for future analysis. Analyzing social network can reveal information about the group or set of groups that have some common goals. A researcher by analyzing social network data (Facebook, Twitter) is trying to visualize the relationship information among the users [15]. In social network domain, researchers [16] discussed the graph network elaborating the degree of distribution, shapes, and some other characteristics using different color schemes. Michael Ley [17] analyzed DBLP dataset to find out various scenarios like the relation of different research institutes, Relationship of an institute within the country, Relationship of researcher and institute outside the country etc.

Citations relation is the key point while calculating impact factor for the journals [18] and ranking research scholars. H-index is an example of such analysis tool [4]. This citation analysis can also be used for ranking universities and other research institutions [19]. Organizations like Scopus, Google and Citeseer utilizes the citation count to rank the research scholar and documents [20] [21]. Some authors highlighted the advantages of in-text citations. Few researcher [22] analyze the relevancy by extracting citation symbols in a different section of the article. Similarly, others [23] [24] analyze research article by author and co-author relation and also with the position of the author in a different research article. Boyack et al. [25] take it to one step further and analyze the research article on the basis of patterns made by citation symbols. Research article's classification into sentiment positive and sentiment negative on the basis of in-text citation is elaborate by Butt et al. [26]. To predict future citation, CiteRank [27] is proposed which integrate publishing time into the random walk model. CiteRank is not that popular because it simply uses citation or publication time to predict future citation. To overcome such concern, Sayyadi and Getoor [28] propose a "FutureRank" model. FutureRank encompasses the properties of CiteRank and also consider citation and author reputation. Like FutureRank model result in high ranking if the author has already published

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2018.2869458, IEEE Transactions on Emerging Topics in Computing

in the high ranking journal. P-Rank[29] consider some metrics of citations, journal reputation, Author reputation and time information and generate the heterogeneous scholarly network to find the research document impact. Wang et al. [30] consider author reputation, citation, journal reputation and time information to find the research document impact.

As scholarly data is increasing day by day, need of new tools and techniques arises to analyze BSD in new ways [31]. ISI impact factor, H-Index and CP & CPP are three popular groups which rank the impact of research articles. The bibliometric indicators are available over the internet to find the rank of the journals and research documents via search engines, for example, Web of Science (WOS), Google Scholar and Scopus. These search engines do not require calculations [5]. CPP calculate impact by taking an average of a number of citations in the document and then generating the result as the average impact of the journal. CPP is calculated by dividing the total number of citations (C) by the total number of articles (P) while neglecting differences in the number of articles published per year. The Hirsch Index (H-index) based on the impact of the author. In H-index, author's impact is based on citations. As the value of H-index goes high, the impact of the author goes higher in research society. With the time h-index also evolved, and new variants were introduced [32-34] time to time. One of the examples is H-Spectrum that is also an indicator of the impact for a research scholar.

Here impact is the capacity of the author to produce impact articles. H-Spectrum can be defined as the distribution of h-indexes for a specific journal in given time for author and co-author [5]. Braun et al. [33] proposed another technique by using H-Spectrum but instead of impact factor, they used "mean citedness" that is identical to CPP. Impact factor uses time frame to calculate impact of a specific journal. Like to calculate impact factor for a given journal we have to consider the average number of citations received by the articles in journals in past two years by total number of articles published in the specific year [35]. There is a misuse of impact factor [36] if it is generalized with dominancy of publishing journals to the single research document, Events or program or even on the basis of discipline included in the publishing journal. It is not necessary that if the publishing journal has a higher impact factor, then author publishing in that journal also has a higher impact factor than other journals or authors. Thomson Reuters calculate and report the impact factor in their annual journal citation report (JRC) for only "ISI-Indexed" by the WoS [37].

3. SCHOLARLY DATA REPRESENTATION USING GRAPHS

As we know, the graphs are the best way to represent any relationship between two entities (such as authors, journals, publishers, articles etc.). The relationships between scholarly entities can better be represented by several graphs, such as author and coauthor relationship, journal to journal citation relationship, author to author citation relationship, author to organization relationship, author to journal relationship, authors to their fields relationship, etc., as shown in Figure 1. These scholarly relationships graph can be used to make any decision related to scholarly entities, such as recommending reviewers based on his field and experience, ranking journals, authors, publishers, etc. In this section, we are describing what scholarly relationships we have used while representing them in the graph to make related decisions, such as finding journals ranking, new impact factor, researcher ranking, and expert reviewers with no conflict of interest.

First, we have used the relationship among all authors in term of citations. Figure 1A shows the relationship in the form of a directed graph (called author-author citation relationship graph-AACR). The weight of the graph represents the number of citations that author A made for author B for all of their publications. For instance, if the weight of the edge $A_i \rightarrow A_j$ is 20 then it means, in all articles of A_i, author A_i cited 20 times the articles of author A_j. The more is the number, the stronger is the relationship, which shows both authors are from the same field. It also shows which author is dominating its inDegree and outDegree towards the corresponding author. The loop on a particular node A_i shows that author A_i cited its own paper either belonging to the same journal or the different one. the weight of the loop shows the number of times author A_i self-cited himself. Which such relationship, we can identify the popular community in the given field. Also, we can find out the most reputed and expert authors by ranking authors in that field. This can be useful for finding out suitable reviewers based on their expertise and rank while excluding self-citations and friends' citations.

Next, the citation relationship among journals is used as depicted as a graph (called journal-journal citation relationship graph-JJCR) in Figure 1B. The nodes are journals, whereas the directed edges $J_i \rightarrow J_k$ show that the journal J_i cited journal J_k. The weight W_{ik} on edge $J_i \rightarrow J_k$ describes the total number of papers the journal J_i cited from J_k. The weighted loop (W_i) on any node J_i shows, all the articles in J_i cited W_i number of articles from same journal J_i. This relationship graph is helpful while working with authors' citation relationship graph to find the journal ranking while avoiding self-journal and self-author citations. Moreover, the publication relationship between authors and journals, as shown in Figure 1C, can be helpful in finding journal ranking. At the time of journal ranking calculation, with this author-journal publication relationship graph (called author-journal publication relationship graph-AJPR). The weight W_{ik} (author A_i published W_{ik} number of articles in journal J_k) can be used to predict the self-author citations made by author A_i for his own papers in other journals; journal ranking mechanism would be described in detail in section 5.

The author coauthor relationship (ACoR) presented in Figure 1D, are used to make active authors community set for a particular field. In this relationship graph, the weight W_{ij} gives a number representing total articles in which the author A_j is coauthor with A_i. Unlike other graphs, this graph does not have any kind of loop on author node A_i (as the author cannot be a coauthor of himself). With the prudent use of this relationship graph, we can make many scholarly data related decisions, such as identifying the strongly co-bounded authors and suitable reviewers while avoiding conflict of interest.

Moreover, sometimes the organization, place, or project information is required where the researchers are working. This type of information might be useful for conflict avoidance while assigning reviewers or it can be used to avoid self-organization citations (even though it is not serious concern of the scholarly community). For these use cases, we built a bipartite author-organization relationship graph (AOPR) as shown in Figure 1E. This graph tracks the record of each author by its weight while

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2018.2869458, IEEE Transactions on Emerging Topics in Computing

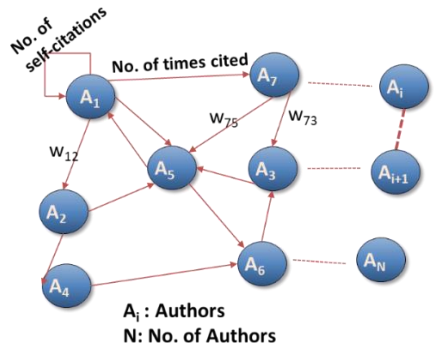


Fig. 1A. Citation relationship graph (AACR graph) among authors.

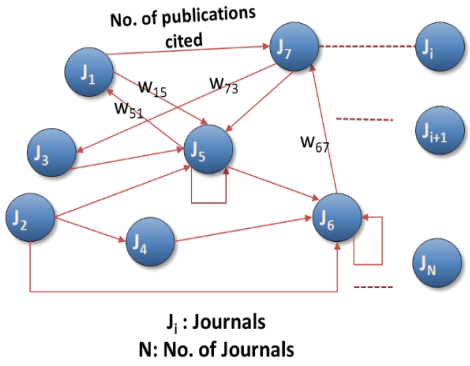


Fig. 1B. Citation relationship among journals (JJCR graph).

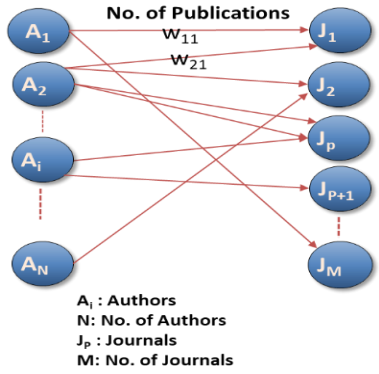


Fig. 1C. Authors and corresponding publication journal relationship (AJPR graph).

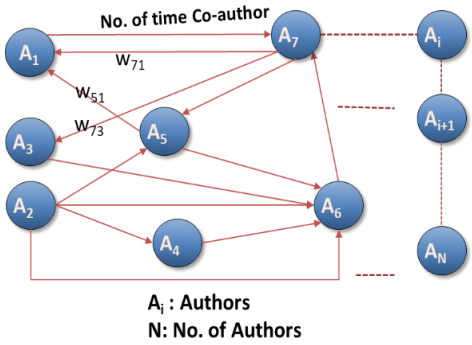


Fig. 1D. Authors and coauthors relationship (ACoR graph)

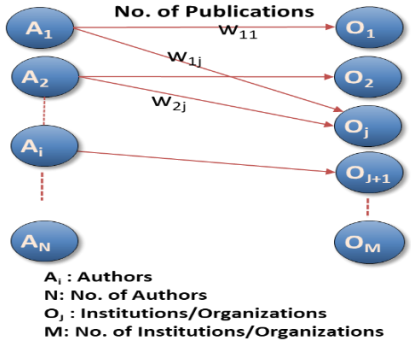


Fig. 1E. Authors and their affiliations relationship (AOAR graph)

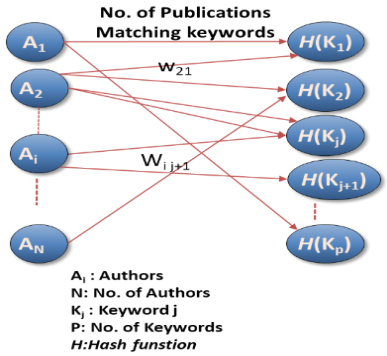


Fig. 1F. Authors and Keyword relationship (AKwR graph).

Fig. 1. Relationship graphs among scholarly entities.

storing the number of publications author A_i published while working at O_j. We create another bipartite graph that we have used to show the authors and their corresponding fields. The fields are taken from the author's papers by extracting the keywords from the title as well as from the keywords section. The hashes of the keywords are taken to store the keywords as nodes to make the searching and processing mechanism optimal. The weight W_{ik} on $A_i \rightarrow H(k_j)$ is the number of times the author A_i used keyword K_j in his paper title or as a keyword. We use this relationship graph (called authors-keywords relationship graph-AKwR), shown in Figure 1F,inr reviewer selection,author'ss ranking and article searching in a particular field.

4. Proposed Big Scholarly Graph Processing Architecture

A simple graph processing can be done with any tool. On the other hand, when we talk about big scholarly data, it generates big graph, which is quite challenging to be processed by traditional tools and techniques. Thus, to handle big scholarly graphs efficiently, we proposed a big graph processing architecture that takes the data from various publishers and repositories and executes various algorithms on that data for scholarly related decision making, as depicted in Figure 2. The proposed system architecture is composed oflayers, i.e. layers i.e., 1) data source layer, graph building layer, graph processing layer, interpretation and decision making layer, and services and application layer. Each of the layers is distinctive in its functionalities. The top one is the data source layer, which does data aggregation from various publishers, such as IEEE, ACM, Elsevier, Google Scholar, IGI global, IET publishers, and other repositories such as Microsoft Academic Graph [38]. This data is in the form of text, articles, database, table, or any other structural form. The real-time scholarly data is also aggregated through Microsoft Cognitive Services Academic Knowledge API [38] from the weekly updating Microsoft Academic Graph repository.

The graph building layer is one of the primary layers of the system, which generates and updates the graphs by taking the incoming data from the data sources. Initially, it creates a new graph, but at later stages, when it finds any new update (through real-time Microsoft Academic Graph), it just updates the graph by either adding a new node, new edge or updating the weights on edge. It uses an efficient searching mechanism, which uses indexing to search particular edge to be updated when required. Graph building layer also increases the efficiency of the system by making the graph to be processed on multiple parallel data nodes simultaneously while dividing the graph into various independent, mutually exclusive parts/subgraphs through the use of Resilient Distributed Datasets (RDD) and Hadoop distributed file system (HDFS). The graph is divided into N subgraphs, i.e., G1, G2, G3, ……., Gn such that G1 ∩ G2 ∩ G3 ∩ …….. ∩ Gn = Φ, as shown in Figure 3. It is more efficient to divide the graphs into subgraph using the cute vertices to achieve high-speed parallelism. However, if the graph G does not have any cut vertex, then it is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2018.2869458, IEEE Transactions on Emerging Topics in Computing

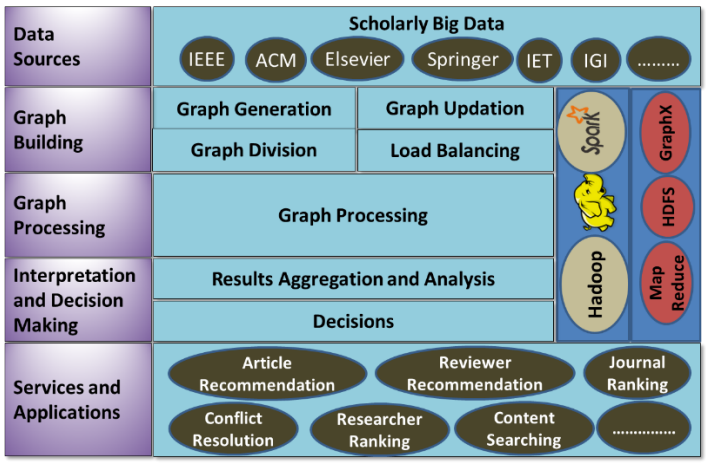


Fig. 2. Proposed layered architecture for big scholarly graph processing

hard to make various components of the graph. The possible number of cuts in the graph can be calculated as $\frac{1}{2}(2^n - 2) = 2^{n-1}-1$ (where n is the number of nodes).

Later, all of the independent subgraphs are sent to the processing server, when processing is required. Therefore, it also performs the load balancing functionality. The processing of the graph is handled by the graph processing layer, which has multiple parallel data nodes to process each individual subgraph. Each of the Subgraph G_i is processed by a distinct data node. Each of the data nodes is equipped with various graph algorithms that run based on the user's request and his needs. At this layer, every node has one output corresponding to each subgraph by the given graph algorithm. The results from all nodes are aggregated for each of the main graph at next layer, i.e., Interpretation and Decision-Making layer, where, after the aggregation, the analyses are performed. Since the processing layer's output is in chunks and each chunk of results corresponds to one subgraph. Therefore, these chunks must be aggregated for final analysis. Finally, at the last layer, the decisions are made based on the analysis results. These results can be journal rankings, new impact factor for each journal, researchers ranking, optimized reviewer search, etc.Overall, the Hadoop ecosystem with Apache Spark and GraphX is used at three intermediate layers. The graphs are generated from the data by using the Spark GraphX tool with the ability of processing large graphs. We use Hadoop ecosystem to achieve the parallel processing of the graph whereas Spark to perform real-time processing on the data. Since MapReduce, the default language of Hadoop is inefficient to process graphs. Therefore, GraphX is the best option to achieve the efficiency while processing graphs. GraphX uses Bulk Synchronous Parallel (BSP) as execution model with distributed system. GraphX also has the vast library of graph processing algorithms. The data is stored in the Hadoop Distributed file system (HDFS) in the form of graphs. The general flow of overall information processing is illustrated in Figure 4.

5. Decision Making using Multilevel Big Scholarly Data

This section provides the details of graphical methods to use the BSD to make a decision. Vital decisions that have considered are (1) expert reviewer selection and avoiding conflict of interest, (2) author ranking computation based on a given field as well as in general, (3) the journal new impact factor (NIF) finding, (4)

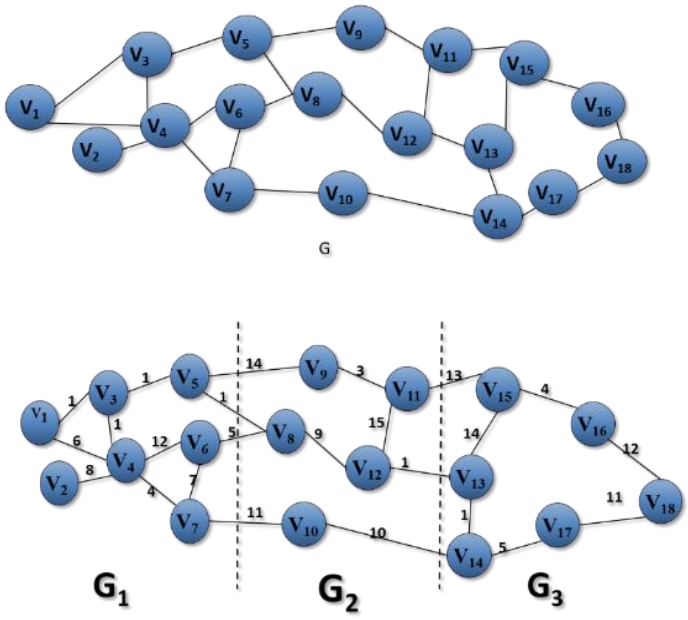


Fig. 3. Division of big scholarly graphs into mutually exclusive subgraphs for efficient processing

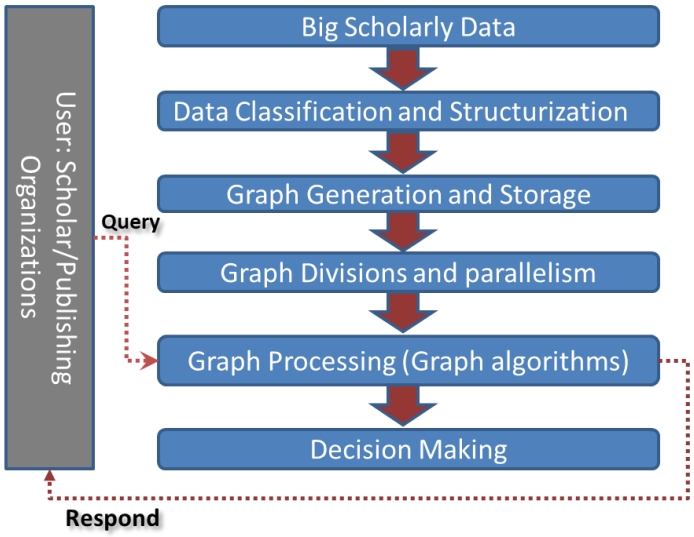


Fig. 4. Flow of big scholarly graph processing

journal ranking computation. For each of these decisions, we used the combination of graphs presented in section 3 to make one general multilevel graph, for example, for perfect reviewer finding, we use 4 levels of the graph. At the top level, author and corresponding keyword relationship (AKWR) graph is used that is deployed to find out the authors working on a particular field/keyword. Next at one down level AKWR graph is connected with the ACOR and AOAR graph. At this level, all the conflicts of interests are identified. The bottom level consists of AACR graph. The analysis of this graph provides the selection of the expert reviewer. Similarly, for researcher ranking AKWR, AACR, ACOR graphs are used whereas, for journal ranking and NIF finding, AACR, JJCR, and AJPR graph are used in multilevel.

5.1 Page Rank

Through the comparison among the entities such as, journals and researchers, we find the key entity by using the modified

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2018.2869458, IEEE Transactions on Emerging Topics in Computing

form of the PageRank algorithm. Page rank algorithm only takes inDegree ($d^-(v)$) and outDegree ($d^+(v)$) of vertices V while ranking any node. Since we are dealing with weighted graph, the weight W_i on an edge A→B considered as in and out degrees as, $(d^+(A))= W_i$ and $(d^-(B))= W_i$. For example, if the weight of an edge is 5, it would be considered 5 times while calculating in and out degrees.

5.2 Expert Reviewer Selection and Conflict of Interest

As already described, for reviewer selection and conflict of interest (CoI) finding, we used the multilevel graph that is built by combining AKWR, ACOR, AACR, and AOAR graphs. For any given paper title *T* and keywords *K*, we proposed the algorithm 1 to find the expert reviewer among all reviewers based on their expertise and fields. After identifying the expert reviewer, the algorithm detects whether there is any conflict of interest between the author and the selected reviewers? It mainly considered the relationship strengths among authors in all graphs. The procedure starts from extracting the keywords from the given title *T*. Once the list of the keywords (we can say that the field of research) are identified, then the *FingReviewer* procedure is called to find out all the potential reviewers in a detailed reviewer list (*RRL*). While selecting the reviewers, initially, all the expert reviewers are identified by matching the authors with the corresponding keywords. If there is an edge from author A_i to title keyword nodes *h(K)* and the weight of an edge is higher than the particular threshold, then it means the author A_i is an expert to the given field. Similarly, all the authors are selected in the same fashion. Next, the subgraph of AACR graph is generated by selecting all the nodes and edges belongs to the related reviewer list (*RRL*). This subgraph is called related author citation graph (RACR). For each researcher in RACR, his co-authors are identified through the processing of ACOR graph. All the citations from the co-authors and the self-citations are ignored in the selected RACR subgraph, as it should not be considered in finding the popularity of the reviewer.

Finally, the page rank algorithm is applied to find the temporary popularity value for all researchers in RRL. In the given field, if the researcher has more publications, the more expertise he has in that field. Thus, finally, reviewers are selected by multiplying the corresponding weight of AKWR graph (as it reflects the number of publication an author has in the given field) to the temporary popularity value. Once the reviewers are finalized, the next phase is to find the conflict of interest (CoI) between the selected reviewers and authors. The procedure reflected in the algorithm 1 is used to find out the CoI. Basically, CoI is detected if and only if the author and the reviewer belong to (or previously belongs to) the same organization, or they have a healthy relationship with each other in an author co-author relationship graph. Two authors A and B have strong relationships with each other if and only if they are farther from each other less than $\int_{MaxHops}$ edges. If they are closer than $\int_{MaxHops}$ we assume that they know each other, as they are co-author or they have some mutual co-authors.

In finding the CoI (or searching any graph node), the navigation/ walk is vital aspect to increase the efficiency of the system. The shorter the walk in the process of searching a node, the faster is the process. Thus, we used decentralized searching mechanism to find a particular node with the desired facility/data by contacting its own friends and friends of its friends using its already built personal friendship network. In the whole searching process, our aim is to reduce the computational cost and select an optimal set of friendships for next link selection. This is done by finding the cluster coefficient of each neighbour node to identify where to move or navigate. The clustering coefficient is calculated for each neighbour node by equation 1 described by watts and strogatz [39]. With the cluster coefficient C_i, the walk probably flows through the edge which has more neighbours or more links i.e., the higher C_i.

$$C_i := \frac{2*(NE_i)}{NK_i*(NK_i-1)} \quad (1)$$

Where NK_i is the total number of nodes directly connected with node i and NE_i represents the number of edges connected to all neighbor i.

Algorithm 1: Perfect Reviewer Finding and Conflict of Interest

Input: AKwR, ACoR, AACR, AOAR Paper title *T*, Authors AL
Output: Reviewers list RL with not Conflict of Interest
Steps:

1. *KWL*[]:= ExtractKeywors(*T*)
2. RRL:= FindReviewer(AL, KWL[],AKwR, ACoR, AACR)
3. For all Authors A_i in AL and all Reviewers R_i in RRL
4. CoI:= FindCoI(A_i, R_i, ACoR, AACR, AOAR)
5. LN: List of Neighbors

FindReviewer(A[], KWL[],AKwR, ACoR, AACR)

$RRL := \{ (A_i, W_i) \mid A_i \in AKwR \wedge A_i \rightarrow H(KWL[]) \wedge W_i := weight(A_i \rightarrow H(KWL[])) \wedge W_i < \lambda \}$
2. $RACR := SubGraph(RRL, AACR)$
3. *LOOP ForEach* Author A_i in *RACR* *Do.*
4. $CoAL_i : (A_i, ACoR)$
5. $RACR := RACR - Exclude(LoopOn(A_i))$
6. $RACR := RAACR - Exclude(CoAL \rightarrow A_i)$
7. *END ForEach LOOP*
8. $TempRanks[\ ] = PageRank(RACR)$
9. *LOOP ForEach* Author A_i in *RACR Do.*
10. $RRL := RRL + Wi * TempRanks[i]$
11. *END ForEach LOOP*
12 *Return RRL*

FindCoI(A, R, ACoR, AOAR)

1. LoO_A:= *OutDegreeNodes*(A, AOAR)
2. LoO_R:= *OutDegreeNodes*(R, AOAR)
3. *ForEach* Element E_A in LoO_A and All element E_R in LoO_R *LOOP.*
4. *IF* $E_A == E_R$ *THEN*
5. *CoI:=True*
6. *Return CoI.*
7. *END IF*
8. *END LOOP*
9. *HopCount* := 1.
10. *LN := Neighbors(ACoR, A)*
11. *IF HopCount* < 1 *THEN*
12. *CoI:= False*
13. *Return CoI*
14. *END IF*
15. *IF R* ∈ *LN THEN*
16. *CoI:= True*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2018.2869458, IEEE Transactions on Emerging Topics in Computing

17. Return CoI
18. END IF
19. IF (HopCount < ∫MaxHops) THEN
20. HopCount++
21. ForEach Node i in LN LOOP
22. Calculate Ci
23. END LOOP
24. LN:= Sort (LN, Ci)
25. Each Node i *in LN PUSH*(Stack[Top])
26. ELSE
27. HopCount- -
28. END IF-Else
29. Curr_node := PoP(Stack[Top])
30. LN := Neighbors(ACoR, Curr_node)
31. Go To Step 11
32. END

5.3 RESEARCHER RANKING

Researcher ranking is important for many reasons. When a scholar or a scientist starts exploring a new research area to find out the typical problems and their solutions, he must discover active scientists who are working, already worked, or popular in that field. Then, he goes for searching and reading their research articles and projects. Thus, findingg the popularity and ranking the researchers is worthwhile to compare scientists belonging to the same field, same career length, or same subject. Even it is good to compare researchers who published in the similar journals. It can be used for analyzing a focused snapshot of a outcomescientist's outcomes.

Therefore, various organizations come up with various researchers' ranking methods, such as h-index and h10-index. H-index frequently used to rank the scientific productivity of a researcher and its impact on the society. Also, itAlso it can be a measure to define journals rankings. The H-index takes the number of total publications and their citations by others as input to give a picture of a particular researcher's performance. For Instance, when an individual published 15 articles and all of them does not have less than 15 citations, it means its h-index is 15. However, significant problems with this approach is that it does not consider self-author citations. Similarly, all other researcher's ranking measurements, including i10-index do not consider journal self-citation (as journal citation is independent). Authors self-citations and the friend's or coauthors' self-citation are loopholes in the existing author ranking approaches as they raised a serious question on the quality measure of all ranking approaches.

The proposed researcher ranking is described by pseudocode presented in algorithm 2. The basic idea of the proposed researcher's ranking is to consider all the citations of the researcher. Also, it depends upon the popularity of the researcher who cited the author whose ranking is to be identified. Moreover, the author collaborators are also identified and all the self-authors citation and citations from the collaborators are excluded while finding the ranking of the author. Algorithm 2 and algorithm 3 presents a step by step procedure to calculate authors ranking based on the given field or without any field respectively. In the algorithms, RAL represents the related researchers list, RAACR represents the related author-author citation relationship graph (subgraph of AACR graph), COAL represents the co-authors lists and λ represents the threshold for number of papers an author published related to the given field (keyword).

Algorithm 2: Finding Researcher Ranks based on a given keyword

Input: AACR, ACoR, AKwR, keyword K
Output: Researchers' Ranks (RR)
Steps:

1. $RAL := \{ (A_i, W_i) \mid A_i \in AKwR \wedge A_i \rightarrow H(K) \wedge W_i := weight(A_i \rightarrow H(K)) \wedge W_i < \lambda \}$ 2. $RAACR := SubGraph(RAL, AACR)$
3. *LOOP ForEach* Author A_i in *RAACR* *Do.*
4. $CoAL_i : (A_i, ACoR)$
5. $RAACR := RAACR - Exclude(LoopOn(A_i))$
6. $RAACR := RAACR - Exclude (CoAL \rightarrow A_i)$
7. *END ForEach LOOP*
8. $TempRanks[\ ] = PageRank(RAACR)$
9. *LOOP ForEach* Author A_i in *RAACR* *Do.*
10. $RR_i := Wi * TempRanks[i]$
11. *END ForEach LOOP*
12 *Return RR*

Algorithm 3: Finding overall Researcher Ranks

Input: AACR, ACoR
Output: Ranks
Steps:

1. *LOOP ForEach* Author A_i in *AACR* *Do.*
4. $CoAL_i : (A_i, ACoR)$
5. $TemGraph := AACR - Exclude(LoopOn(A_i))$
6. $TemGraph := TemGraph - Exclude (CoAL \rightarrow A_i)$
7. *END ForEach LOOP*
8. $RR = PageRank(TemGraph)$
12 *Return RR*

5.4 JOURNAL NEW IMPACT FACTOR(NIF)

Journal impact factor is essential to check the reputation of the journal. Numerous metrics are defined for journals ranking and reputations based on citations they have. *Thomas Router* is one of the most popular impact factor measurement that uses thecitationsf the number of total citation and the number of total publications. Also, *Eigenfactor* is another measurement that also considers the number of citations, but gives more weight to citations from reputed journal and less weight to the low ranked journals citations. The more weight for highly ranked journal contributes extra to the eigenvector as compare to the low quality journals. Furthermore, the one called *SCImago Journal Rank* is a metric for scholarly journals to measure the their scientific influence by considering both total number of journal's citations and the number of citations coming from high ranked journals. Whereas, *Altmetrics* rates journals by the overall references posted on academic social media sites [40].

Even though actual impact factor measurements excludes journal self-citations, but still there are flaws in the techniques. These days, some authors self-cite his own papers from other journals, even though papers are not related. Moreover, few other researchers also cite their friends and collaborator's papers. These flaws affect the quality of the journal and these citations should not be counted while measuring the impact factor. In our new impact factor finding technique, we avoid all of such flaws by using the probabilistic measures and multilevel

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2018.2869458, IEEE Transactions on Emerging Topics in Computing

graph constructed from AACR, JJCR, and ACOR graphs. We avoided self-journal citations, self-authors citation, and collaborators and friends citations. We Identified the probability of self-author citation (PR (SC)) by Equation 2 that provides the estimation that how much is the probability that a citation is a self-author citation (an author cited his own paper (can be from another journal)).

$$PR\ (SC) = \frac{TASC}{TC} \quad (2)$$

Where $TASC$ is total authors self-citations, and TC is the Total Citation count. $TASC$ can be measured using AACR graph. Whereas, TC can be calculated from JJCR or AACR.

Next from the self-citation probability PR(SC), we predict the number of possible author Self-citations in a particular journal K (ASCJ). The Eq. 3 give the estimation of ASCJ for any journal K.

$$ASCJ_K = \sum_{i=0}^{NAJ} NP_i - (No.of.LoopsOn(A_i, AACR) + PR\ (SC)) \quad (3)$$

Where NP_i is the number of publications of an author Ai in the journal J_K. NAJ is the number of authors who published in the journal J_K. $No.of.SelfCitation(A_i)$ is the function to calculate the number of self-authors citation.

Finally the new impact factor of Journal K (NIF_K) is calculated by Eq. 4 by excluding any type of self-citations.

$$NIF_K = \frac{TJC_K - (TJSC_K + SACJ_K)}{TJ_KP} \quad (4)$$

Whereas TJC_K represents the total citations of a particular Journal K, $TJSC_K$ represents the total Journal Self-Citations of a particular Journal K, $SACJ_K$ is self-author's citation in journal K, and TJ_KP describes the total publication of a particular Journal K.

5.5 Journals Ranking

Journal ranking is widely used in academic circles for the evaluation of an academic journal's impact and quality. In finding the journal ranking, we only considered the journal new impact factor (NIF) but also the reputation of other journals who cited the journal whose rank is to be calculated. Alike existing techniques, we have also considered the number of citations while excluding self-journal citations. The journal rank graph is calculated excluding self-journal citation by equation 5, and the temporary ranks are identified by page rank (PR) as equation 6. Page rank considers the popularity of the citing journal as well while calculating the journal ranks. Final journal ranking is measured by multiplying NIF with journal temporary rank by equation 7.

$$JournalRankGraph = JJCR - Exclude(LoopOn\ JJCR) \quad (5)$$

$$TempRanks = PageRank(JournalRankGraph) \quad (6)$$

$$JournalRanks[J_i,\ rank] = NIF_i * TempRanks\ (J_i) \quad (7)$$

Where $1 < i <$ No. of Journals

The proposed journal impact factor, journal ranking, researcher ranking, reviewer selection and conflict avoidance mechanism reflect the needs of the scholarly communities to improve the quality of scholarly data as well as the journal. It also guarantees the fair measuring mechanism of journal ranking, impact factor, and researchers ranking, while removing all the flaws exists in the community.

6. System Implementation and Evaluation

6.1 Experimental Setup

Selection of right platform for big data analysis is a bit tricky. We choose Apache Spark, because of its real-time analysis enactment. Ubuntu version 16.04 is selected to host Spark. To perform parallel processing, the Hadoop ecosystem is used under the Spark plateform. Once all spark's parameters are correctly configured with Ubuntu, decision have to be made for programming languages to interact with the Hadoop. In our experimental setup, we use Core i5 processors with 16 GB of RAM. Specific configuration is made to tune Apache Spark for a better result. Apache Spark's driver, executor and core configuration can be modified besides its default settings. For our experiment we assign 10 GB of memory for Apache Spark's driver and executor. Four cores are assigned to spark for parallel execution of spark data nodes on Ubuntu terminal. The parallel execution of the graph is achieved through a cut vertex mechanism. Initially, we check whether the graph has a cut vertex or not, if it has then it can be separate.

6.2 Datasets

As several scenarios are devised to find specific facts and figures, thus, multi-datasets are required to perform experiments, analysis, and evaluation. Firstly, AMinor [41] dataset named as "CoAuthor" and Google Scholar [42] dataset are taken that present author-coauthor relationship, scholar-citation relationship, and paper-to-paper citations relationship. Secondly, other useful datasets showing various scholarly relations from AMinor are considered. These datasets are presented in Table 1. In addition, for evaluation purpose, 260 MB citation dataset is also used that is generated by Sugiyama and Kan [43] for scholarly paper recommendation. The dataset covers 100,531 papers and 50 authors while storing IDs of papers, citation information, and reference information about each candidate papers from ACL Anthology Reference Corpus. Thirdly, due to the importance of Scopus community, we have also practiced a huge dataset of 26GB (19GB +07GB) named as Open Research Corpus datasets [44]. It contains all the information including paper id, title, Abstract, keywords, authors, citations (inCitations, outCitations), journal details (name, volume, pages) of over 20 million published research papers plus 7 millions papers in Computer Science, Neuroscience, and Biomedical fields. Finally, for the most updated and real-time scholarly data, we have taken the Microsoft Academic Graph (MAG) [38] with 166,192,182 papers. The dataset is updated weekly via Microsoft Cognitive Services Academic Knowledge API by constructing the heterogeneous graph containing scientific publication records and relationships among those publications, authors, institutions, journals, and conferences.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2018.2869458, IEEE Transactions on Emerging Topics in Computing

6.3 System Evaluation

Table 2 shows the summary of spark implementation details for two basic tasks, i.e., Journal ranking using PageRank and Keyword searching using the greedy approach on google scholar and coauthor (AMiser) dataset. The google scholar dataset of size 43MB has 82937 nodes and 148116 edges, whereas coauthor dataset is little bigger of size 74MB that has 1560640 nodes and 4258946 edges. The table shows the number of the parallel RDD division of the graph dataset. The more RDD blocks, the more parallelism can be achieved. Also, if either the action or the dataset is more complicated then we need more RDDs. The table also summarizes the use of memory for the dataset for the particular proposed algorithm. Moreover, the overall algorithm is divided into multiple jobs, which are further divided into multiple tasks and stages to achieve the multilevel parallelism. The huge job can be divided into small parallel processing jobs to increase efficiency. Since the Coauthor is bigger than google scholar dataset, it is using more RDDs, More task divisions into jobs and stages, more memory usage, and more processing time. Similarly, journal ranking is more complicated procedure than the keyword searching, thus it has the same effects.

Furthermore, we are dealing with big scholarly data. Thus, we evaluated our proposed system while considering its efficiency. We considered the processing time with respect to increase in number of nodes and number of edges. Figure 5 shows the processing time of NIF finding and journal ranking corresponding to the number of nodes. Whereas Figure 6 shows the time spent on keywords searching and author ranking corresponding to the increasing number of edges with constant number of nodes i.e., 10K. In all of these cases, with a very huge number of increase in edges and edges there is a very little increase in the processing time. However, the increase in processing time is more corresponding to the number of nodes than a number of edges.

Almost results are observed with other algorithms as we have seen in the cases of NIF finding, journal ranking, keyword searching, and authors ranking. Figure 7 shows the processing time (ms) consumed by the proposed system while selecting expert review and identifying a conflict of interest. You can see, with more than 50 thousand nodes, the processing time is quite lower, i.e., 1600ms in case of conflict of interest and 2500ms in the case of expert reviewer selection. It is also obvious that conflict of interest takes very less time than the reviewer

Table 1. Datasets

Dataset Name	Nodes	Edges and Description
Citation	1572277 papers	2084019 citation relationships
Academic Social Network	2,092,356 papers. 1,712,433 authors	8,024,869 citation relationships. 4,258,615 coauthor relationships
Topic-coauthor	640134 authors of 8 topics	1554643 coauthor relationships
Dynamic coauthor	1629217 authors	2623832 coauthor relationships
Expert Finding	1781 experts	13 topics
Topic model results-Arnetminer dataset	Top 1000000 papers and authors	200 topics
Coauthor	1560640 authors	4258946 coauthor relationships
Disambiguation	110 authors	Affiliations

Table 2. Spark GraphX implementation Details.

Datasets	**Google Scholar Dataset (43MB)**		**Coauthor (74MB)**	
Actions	Journal Ranking	Keyword Searching	Journal Ranking	Keyword Searching
Total RDD blocks	45	7	62	30
Total Tasks	282	8	807	48
Stages	141	4	269	10
Jobs	36	2	68	10
Storage Memory (MB)	21.7	6.3	1400	952
Processing Time (sec)	49	3	636	43

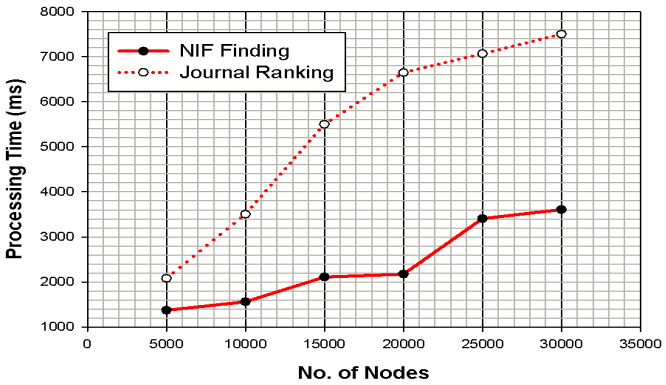
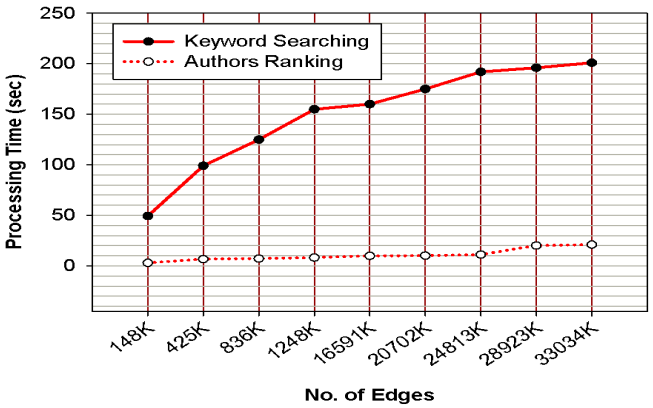


Fig. 5. Processing time of new impact factor finding mechanism and journal ranking approach corresponding to increasing in nodes



This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2018.2869458, IEEE Transactions on Emerging Topics in Computing

Fig. 6. Processing time of Keyword Searching procedure and Authors ranking procedure corresponding to increasing in edges

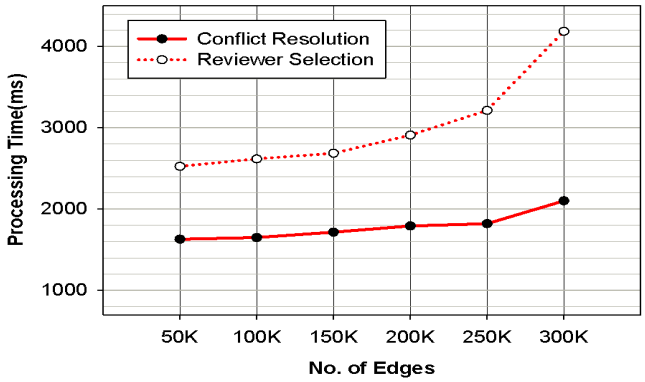


Fig. 7. Processing time of Conflict of Interest procedure and Reviewer Selection procedure corresponding to increasing in edges.

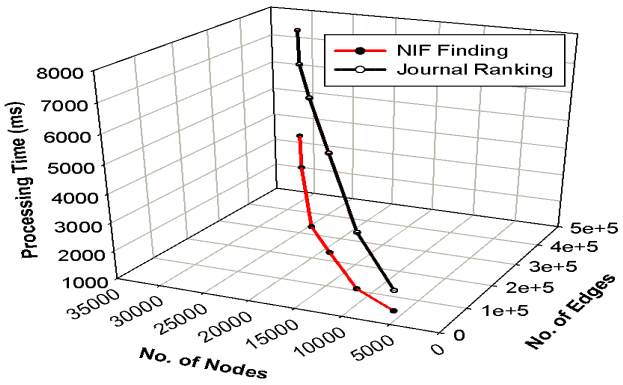


Fig. 8. Processing time of Conflict of Interest procedure and Reviewer Selection procedure corresponding to increasing edges and nodes

selection procedure. This is because of a short number of stages performed by conflict of interest procedure. For conflict of interest identification, we use efficient navigation/walking mechanism. Moreover, we just need to traverse a very short portion of the whole graph while detecting CoI. Thus, CoI takes very short time as compared to expert reviewer selection. Also, in both cases, there is very short increase in processing time with a very substantial increase in both, the number of nodes and number of edges, which is shown in Figure 8.

With these results, it is obvious that the proposed system is quite efficient and meet the needs of the scholars' community. With our best knowledge, the proposed system with the unique algorithms are novel and better than existing approaches

7. Conclusion

The proposed graph-based analysis mechanisms performed well to make decisions for the defined four critical problems that are raised by the research community. By analyzing the graph, the proposed mechanism selects and recommends suitable reviewers for a submitted research article based on researchers' expertise and their reputation in that certain field while this mechanism successfully avoids the conflict of interest. The journal ranking mechanism generates promising results while computing new impact factor and relative ranking by using a modified version of traditional page ranking algorithm and excluding self-authors citations as well as self-journal citations. Researchers ranking is also highlighted, as it is important for various motives, which is calculated on the basis of expert's field, citation count, and a number of publications by avoiding all loopholes. We proposed an architecture to process large graph generated by scholarly data on web. The proposed architecture uses the parallel processing mechanism of the Hadoop ecosystem over the real-time analysis tool, i.e., Apache Spark with GraphX that efficiently processed the big graphs generated by a huge number of scholarly related relationships.

Acknowledgement

This study was supported by the National Research Foundation of Korea(NRF) grant funded by the Korean Government (NRF-2017R1C1B5017464).

References

1. Xia, Feng, et al. "Big Scholarly Data: A Survey." IEEE Transactions on Big Data 3.1 (2017): 18-35.
2. M. Khabsa and C. L. Giles, "The number of scholarly documents on the public web," PloS One, vol. 9, no. 5, 2014, Art. no. e93949.
3. E. Rogers, Diffusion of Innovations, New York:Free Press, 2003.
4. J. E. Hirsch, "An index to quantify an individual's scientific research output", Proc. Nat. Acad. Sci. USA, vol. 102, no. 6, pp. 16569-16572, Nov. 2005
5. F. Franceschini, D. Maisano, "2010-b. The hirsch spectrum: A novel tool for analysing scientific journals", J. Inf., vol. 4, no. 1, pp. 64-73.
6. W. R. Stone, S. J. Wijnholds, and P. Wilkinson, "The Academic Publishing Process: From Manuscript to Publication," Radio Science Bulletin, No. 357, June 2016, pp. 61-68.
7. Commission on Publication Ethics, "COPE Ethical Guidelinesfor PeerReviewers," available athttp://publicationethics. org/files/Peer%20review%20guidelines_O.pdf.
8. S. J. Wijnholds, W. R. Stone and P. Wilkinson, "Early career representative column: Ethical issues in academic publishing," in URSI Radio Science Bulletin, vol. 89, no. 4, pp. 53-59, Dec. 2016. doi: 10.23919/URSIRSB.2016.7910007
9. Zhou, Liang. "QoE-driven delay announcement for cloud mobile media." IEEE Transactions on Circuits and Systems for Video Technology 27, no. 1 (2017): 84-94.
10. Zhou, Liang. "On data-driven delay estimation for media cloud." IEEE Transactions on Multimedia 18, no. 5 (2016): 905-915.
11. Rathore, M. Mazhar, et al. "Efficient graph-oriented smart transportation using internet of things generated big data." Signal-Image Technology & Internet-Based Systems (SITIS), 2015 11th International Conference on. IEEE, 2015.
12. Rathore, M. Mazhar, Awais Ahmad, Anand Paul, and Uthra Kunathur Thikshaja. "Exploiting real-time big data to empower smart transportation using big graphs." In Region 10 Symposium (TENSYMP), 2016 IEEE, pp. 135-139. IEEE, 2016.
13. Abdul, Rehman, Muhammad Mazhar Ullah Rathore, Anand Paul, Faisal Saeed, and Raja Wasim Ahmad. "Vehicular Traffic Optimization and Even Distribution using Ant Colony in Smart City Environment." IET Intelligent Transport Systems (2018).
14. Rathore, M. Mazhar, Hojae Son, Awais Ahmad, and Anand Paul. "Real-time video processing for traffic control in smart city using Hadoop ecosystem with GPUs." Soft Computing 22, no. 5 (2018): 1533-1544.
15. Elmacioglu Ergin, Dongwon Lee, "On six degrees of separation in

DBLP-DB and more", ACM SIGMOD Record 34, vol. 2, pp. 33-40, 2005.
16. John C. Paolillo, "Analyzing linguistic variation: Statistical models and methods", Center for the Study of Language and Inf, 2002.
17. Mark EJ. Newman, "The structure of scientific collaboration networks", Proceedings of the National Academy of Sciences 98, vol. 2, pp. 404-409, 2001.
18. E. Garfield, "Citation analysis as a tool in journal evaluation", Amer. Assoc. Advancement Sci., vol. 178, pp. 471-479, Nov. 1972.
19. A. H. Goodall, "Should top universities be led by top researchers and are they? A citations analysis", J. Doc., vol. 62, no. 3, pp. 388-411, May 2006.
20. J. Beel, B. Gipp, "Google Scholar's ranking algorithm: The impact of citation counts (an empirical study)", Proc. 3rd Int. Conf. IEEE Res. Challenges Inf. Sci., pp. 439-446, Apr. 2009.
21. C. L. Giles, K. D. Bollacker, S. Lawrence, "CiteSeer: An automatic citation indexing system", Proc. 3rd Conf. Digit. Library, pp. 89-98, May 1998.
22. Teufel, Simone, and Min-Yen Kan. "Robust argumentative zoning for sensemaking in scholarly documents." Advanced Language Technologies for Digital Libraries. Springer, Berlin, Heidelberg, 2011. 154-170.
23. B. Gipp, J. Beel, "Citation proximity analysis (CPA)—A new approach for identifying related work based on co-citation analysis", Proc. 12th Int. Conf. Scientometrics Inform., vol. 2, pp. 571-575, Jul. 2009.
24. S. Liu, C. Chen, "The effects of co-citation proximity on co-citation analysis", Proc. ISSI, pp. 474-484, 2011.
25. S. Liu, C. Chen, "The effects of co-citation proximity on co-citation analysis", Proc. ISSI, pp. 474-484, 2011.
26. Butt, Bilal Hayat, et al. "Classification of research citations (CRC)." arXiv preprint arXiv:1506.08966 (2015), [online] Available: https://arxiv.org/abs/1506.08966.
27. P. D. Batista, M. G. Campiteli, O. Kinouchi, A. S. Martinez, "Is it possible to compare researchers with different scientific interests?", Scientometrics, vol. 68, no. 1, pp. 179-189, 2006.
28. T. Braun, W. Glänzel, A. Schubert, "A Hirsch-type index for journals", Scientist, vol. 69, no. 1, pp. 169-173, 2006.
29. D. Walker, H. Xie, K.-K. Yan, and S. Maslov, "Ranking scientific publications using a model of network traffic," J. Statistical Mech.: Theory Experiment, vol. 2007, no. 6, 2007, Art. no. P06010.
30. H. Sayyadi and L. Getoor, "FutureRank: Ranking scientific articles by predicting their future PageRank," in Proc. SIAM Int. Conf. Data Mining, 2009, pp. 533–544.
31. E. Yan, Y. Ding, and C. R. Sugimoto, "P-rank: An indicator measuring prestige in heterogeneous scholarly networks," J. Amer. Soc. Inf. Sci. Technol., vol. 62, no. 3, pp. 467–477, 2011.
32. Y. Wang, Y. Tong, and M. Zeng, "Ranking scientific articles by exploiting citations, authors, journals, and time information," in Proc. 27th AAAI Conf. Artif. Intell., 2013, pp. 933–939.
33. Y. Dong, R. A. Johnson, and N. V. Chawla, "Can scientific impact be predicted?" IEEE Trans. Big Data, vol. 2, no. 1, pp. 18– 30, Jan.-Mar. 2016.
34. J. BiHui, L. LiMing, R. Rousseau, L. Egghe, "The R- and AR-indices: Complementing the h-index", Chinese Sci. Bull., vol. 52, no. 6, pp. 855-963, 2007.
35. E. Garfield, "Agony and the ecstasy—the history and meaning of the impact factor", Int. Congr. Peer Rev. Biomed. Publ. Chicago IL USA, 2005-Sep. 16.
36. M. Amin, M. Mabe, "Impact factors: Use and abuse." Elsevier Science Perspectives in Publishing, Oct. 2000.
37. Thomsonreuters, "Journal_Citation_Reports", [Online].Available: www.thomsonreuters.com/products_services/scientific/Journal_Citation_Reports, [ACCESSED ON: June 04, 2018].
38. Sinha, Arnab, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-june Paul Hsu, and Kuansan Wang. "An overview of microsoft academic service (mas) and applications." In Proceedings of the 24th international conference on world wide web, pp. 243-246. ACM, 2015.
39. D. J. Watts and S. H. Strogatz, "Collective dynamics of smallworldnetworks," Nature, vol. 393, no. 6684, pp. 440–442, 1998.
40. lhoori, Hamed; Furuta, Richard (2013). "Can Social Reference Management Systems Predict a Ranking of Scholarly Venues?". Research and Advanced Technology for Digital Libraries. Lecture Notes in Computer Science. 8092: 138–143. doi:10.1007/978-3-642-40501-3_14. ISBN 978-3-642-40500-6.
41. AMiner Dataset. [Available at]. https://aminer.org/data [accessed on Nov. 27, 2017]
42. Google Scholar citation relations.[Available at]. http://www3.cs.stonybrook.edu/~leman/data/gscholar.db [accessed on Nov. 27, 2017]
43. Kazunari Sugiyama and Min-Yen Kan: "Exploiting Potential Citation Papers in Scholarly Paper Recommendation," The 13th ACM/IEEE Joint Conference on Digital Libraries (JCDL 2013), pp.153-162, Indianapolis, Indiana, USA, July 22-26, 2013. Australia, June 21-25, 2010. [Download]: http://www.comp.nus.edu.sg/~sugiyama/SchPaperRecData.html [Accessed on Nov. 27, 2017].
44. Open Research Corpus datasets. [Available at]. http://labs.semanticscholar.org/corpus/ [Accessed on: Nov 27, 2017].

Muhammad Mazhar Ullah Rathore received his Master's degree in Computer and Communication Security from the National University of Sciences and Technology, Pakistan in 2012. Currently, he is pursuing his Ph.D. with Dr. Anand Paul at Kyungpook National University, Daegu, South Korea. His research interests include Big Data Analytics, Internet of Things, Smart Systems, Network Traffic Analysis and Monitoring, Remote Sensing, Smart City, Urban Planning, Intrusion Detection, and Computer and Network Security. He is an IEEE and ACM student member. He got the best project/paper award in Qualcomm Innovation Award 2016 at Kyungpook National University, Korea for his paper "IoT-Based Smart City Development using Big Data Analytical Approach". He is also a nominee of Best Project Award in 2015 IEEE Communications Society Student Competition for his project "IoT based Smart City". He is serving as a reviewer for various IEEE, ACM, Springer, and Elsevier journals.

Malik Junaid Jami Gul received his Master's degree in computer science and Information Technology, Pakistan in 2015. Currently, he is pursuing his Ph.D. with Dr. Anand Paul at Kyungpook National University, Daegu, South Korea. His research interests include Big Data Analytics, Internet of Things, Smart Systems, Network Traffic Analysis and Monitoring, Smart City, Operating system security, Intrusion Detection, and Computer and Network Security.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2018.2869458, IEEE Transactions on Emerging Topics in Computing

Anand Paul received the Ph.D. degree in electrical engineering from the National Cheng Kung University, Tainan, Taiwan, in 2010. He is currently working as an Associate Professor with the School of Computer Science and Engineering, Kyungpook National University, Daegu, Korea. He is a delegate representing Korea for M2M focus group and for MPEG. His research interests include algorithm and architecture re- configurable embedded computing. Prof. Paul has Guest Edited various international journals and he is also part of the Editorial Team for Journal of Platform Technology and Cyber Physical Systems. He serves as a Reviewer for various IEEE/IET journals. He is the track Chair for smart human computer interaction in ACMSAC 2015, 2014. He was the recipient of the Outstanding International Student Scholarship Award in 20042010, the Best Paper Award in National Computer Symposium, Taipei, Taiwan, in 2009, and UWSS 2015, in Beijing, China. He is also IEEE Senior Member.

Ashraf Ali Khan received his B.E. degree in Electronics Engineering from National University of Sciences and Technology (NUST), Islamabad, Pakistan, in 2012, and his M.S. combined Ph.D. degree in Energy Engineering from Kyungpook National University, Korea. He is currently working as a postdoctoral research fellow and lecturer in the University of British Columbia, Canada. His current research interests include high efficiency and high reliability power converters, magnetics, grid connected inverters, multilevel converter systems and smart grid and systems.

Raja Wasim Ahmad is an Assistant professor at COMSATS Institute of Information Technology, Pakistan. He did his PhD in Computer Science from University of Malaya under Bright Spark Scholarship program. He started his carrier as a computer student back in 2003 by choosing computer science as a major during under-graduation course from University of Azad Jammu & Kashmir, Muzaffarabad. In addition, he did his masters from COMSATS Institute of Information Technology, Abbottabad, Pakistan under "COMSATS Merit Scholarship" program. His research interests include mobile application energy profiling, energy efficient computational offloading, cloud resource allocation, VM migration, Network performance, Big data analytics, Application's QoS on low bandwidth networks, and energy efficient cloud data centers

Joel J. P. C. Rodrigues [S01, M06, SM06] is a professor at the National Institute of Telecommunications (Inatel), Brazil and senior researcher at IT, Portugal. He has been professor at UBI, Portugal and visiting professor at UNIFOR. He is the leader of Net- GNA Research Group, the President of the scientific council at ParkUrbis-Covilh Science and Technology Park, the Past-Chair of the IEEE ComSoc TCs on eHealth and on Communications Software, Steering Committee member of the IEEE Life Sciences Technical Community. He is the editor-in-chief of three international journals and editorial board member of several journals. He has authored or co-authored over 500 papers in refereed international journals and conferences, 3 books, and 2 patents.

Dr. Spiros Bakiras received his B.S. degree in Electrical and Computer Engineering from the National Technical University of Athens in 1993, his M.S. degree in Telematics from the University of Surrey in 1994, and his Ph.D. degree in Electrical Engineering from the University of Southern California in 2000. Currently, he is an associate professor in the Information and Computing Technology Division of Hamad Bin Khalifa University. Before that, he held academic positions at various institutions, including Michigan Tech, the City University of New York, and the Hong Kong University of Science and Technology. His current research interests include secure data management, applied cryptography, mobile computing, and spatiotemporal databases. He is a member of the ACM and a recipient of the U.S. National Science Foundation (NSF) CAREER award.