Ranking of Academic Papers

Department of Computer Science, Stony Brook University

***Abstract*— The purpose of this project is to construct a ranking metric to evaluate academic papers and researchers using available data in citation network dataset. One of the popular metric h-index is not very reliable when it comes to ranking authors. We have come up with a new metric named P^2. This metric is created using the normalized pagerank of individual authors as well as the normalized pageranks of the papers they have authored. Because P^2 uses normalized pagerank scores, it is possible to rank authors and papers even across different domains which is not possible if we use h-index. Also, we have come up with a metric named FutureRank to predict their popularity in future.**

I INTRODUCTION

The citation is a way of acknowledging the related work of others to the topic of discussion. These citations can be used to devise a metric to find the impact of a paper/article. One of the most basic citation metrics is how often an academic paper/article is referenced in the other papers, articles, books or any other sources. One can infer that higher the number of citation, the more important is the paper. But there are some problems with this metric, as it does not take into account the factor such as discipline, the number of people working in that area, coverage of the paper. There has been a lot of research in this area to find a universal citation index. One of the very first citation Indices that gain popularity was Science Citation Index(SCI) and Social Science Citation Index(SSCI), both were originally produced by Institute of Scientific Information and created by Eugene Garfield [1].

In recent years, many systems are developed to quantify the cumulative impact and relevance of an individual's scientific research work. In a world where the number of researches are ever increasing, such quantification system is often needed for evaluation and comparison purposes. One of the popular such systems is Hirch-index, also known as h-index. h-index is defined as the number of papers with citation number $\geq$ h. A scientist has index h if h of his or her N papers have at least h citations each and the other N - h papers have less than h citations each. For comparing two researchers, if their total number of papers or their total number of citations is very different, h-index allows to compare them in terms of their overall scientific impact. Conversely, comparing two individuals having almost same scientific age with a similar number of papers or total citation count and very different h-values, the one with the higher h is likely to be the more accomplished.[2]

The current ranking system based on either the number of papers, citations or by the Hirsch index does not consider the problem of distributing authorship among authors in multi-author publications. Irrespective of if the paper is single-author or multi-author, publications contribute to the publication record of a researcher equally. This full counting scheme is apparently unfair and causes unjust disproportions, in particular if ranked researchers have distinctly different collaboration profiles. In a long-term prospect, the research itself is somewhat dependent on research evaluation and the definition of ranking criteria. As a result, researchers try to improve their ranking by complying with the presently accepted criteria and hence the number of citations is being increased by self-citations, and the number of published papers and the h-index are rising over time by the inflated number of multi-author publications and the number of authors.[3]

The advantage of the h-index is that it is insensitive to the tail of lowly cited papers and sensitive to the level of the highly cited papers. But once a paper belongs to the top h class, it is completely unimportant whether these papers continued to be cited or not. If cited, it is unimportant how many more citations they receive. The measure should indicate the overall quality of a scientist with the performance of top papers and their citation count even they belong to the top class. [4] Thus g-index is introduced as an improvement of the h-index. In g-index, the set of articles are ranked in the decreasing order of the number of citations they received. The g-index is the largest number such that top g papers in the sorted list have received a total of g^2 citations together. [5]

Based on the foundation laid by the h-index, one new system was developed, the R- and AR-indices. They are used in combination with the h-index to eliminate the disadvantages of h-index. The R-index measures the h-cores citation intensity where h-core is all the publications ranked between rank 1 and rank h. AR index takes the age of publications into account. This allows for an index that can actually increase and decrease over time.[6]

The h-index does not take the exact number of citations of articles included in the h-core into account, A-index tries to correct this fact. This index is simply defined as the average number of citations received by the publications included in the h-core and hence the name A(for average).[6]

The A-index has one limitation though. Suppose there are two scientists, S1 and S2. scientist S1 has published 20 articles, one cited 10 times and all other ones just once. Scientist S2 has published 30 articles, one cited 10 times and all other ones exactly twice. Clearly, scientist S2 is the better one. This is expressed by their h-indices which are 1 for S1 and 2 for S2. But when you compare their A-indices, it is 10 for S1 and 6 for S2. The better scientist is punished for having a higher h-index, as the A-index involves a division by h. This problem can easily be solved by simply taking the sum, or, the square root of the sum. Taking the square root has the advantage of leading to indicator values which are not very high and of the same dimension as the A-index. This new

index is referred as R-index(root).[6]

In order to overcome the problem of the h-index that it may never decrease, an age-dependent adaptation of the R-index which is denoted by AR is proposed. If there are several publications with exactly h citations then the most recent ones are included in the h-core. The advantage of the AR-index is, besides taking the actual number of citations into account, it also uses the age of the publications. In this way, the h-index is complemented by an index that can actually decrease. The pair (h, AR) is proposed as a meaningful indicator for research evaluation.[6]

To measure the importance of the authors and papers, we have developed a metric which is based on page rank. When a paper cites another paper, it increases the importance of the cited paper and the authors who contributed to the cited papers. PageRank thinks of links as votes, where a page linking to another page is casting a vote. The more often a particular paper is cited by other papers, the more important that paper is deemed. But our index gives importance to the co-authors and the paper in ranking them, which overcomes the limitations of h-index. We will explain this in detail in section 5.

II DATASETS

We have used the DBLP citation network version 10 [7]. It contains a total of 3,079,007 papers and 25,166,994 citation relationships. The dataset is divided into 4 files. First three file contains 1 million papers and the last one contains the rest of the papers. Each paper is represented in the JSON format. The JSON schema has 8 fields which are as following:

- **Id**: Unique Id of the paper
- **Title**: Title of the paper
- **Author**: List of papers authors
- **Venue**: Conference name where paper is presented
- **Year**: Year of publication
- **Number of citations**: Total number of papers that has cited this paper
- **References**: List of paper Id that has been cited by this paper
- **Abstract**: Abstract of the paper

The problem with the above dataset is that they have not mentioned the domain of the paper. As our final task to predict the top authors of each domain we have to cluster our authors by their domains. For that, we are using the Aminor-Author dataset [8]. It contains a total of 1,712,433 authors. Each author is represented in the JSON format. The JSON schema has the 9 fields which are as follows:

- **#index**: index id of this author
- **#n** : name of the author
- **#a**: affiliations of the author
- **#pc**: the number of published papers of this author
- **#cn**: the total number of citations of this author
- **#hi**: the H-index of this author
- **#pi**: the P-index with equal A-index of this author
- **#upi**: the P-index with unequal A-index of this author
- **#t**: extracted key-terms of this author (separated by semi-colons)

From the above-given fields we have used the name of the author, the affiliation of the author and the extracted key-terms of the author for our purpose.

A. Preprocessing of the datasets

For the clean up task on the DBLP dataset we removed the column of venue and abstract as we have not used them, removed the papers that have NaN for the value either Id, Author or Reference field and filled the number of citation field with zero for the papers that have NaN in the number of citation field. Here we are working on a subset of DBLP dataset as running pagerank and HITS algorithm on whole dataset is taking too much time.

Using above dataset as a base we generate a new dataset for the authors. Here a unique Id is given to all the authors. Each author row contains the following information:

- **Id**: A unique id given to all the authors.
- **name**: the name of author
- **co-author Ids**: This feature is a list of all the author with whom the author has published papers. We iterate over the whole citation network dataset and appended all the authors of a paper as the co-author of each other.
- **total_citations**: Total number of citation by all the papers that the author has written
- **total_papers**: Total number of papers the authors have published.
- **h-index**: Along with the above information a separate list is generated that contains pairs of the paper-Ids and their respective number of citations for each author. Using this data, h-index is calculated as **number of papers(h) with a citation number h** for this author.

III METHODS

A. Baseline Model

We have created a baseline model for ranking authors. In this baseline model, we have ranked the authors based on their total number of citations, but divided equally among co-authors. The Equation is as follows:

$$Baseline\_index = \sum_{i=1}^{total\_publications} \frac{total\_number\_of\_citations}{number\_of\_co\_authors} \quad (1)$$

For example, if a paper has 50 citations and it has 5 authors, we will give 10 points to each author. there are some advantages of this metric. H-index does not consider the citations for some of the papers, which, we believe should not be the case. For example, if an author has h-index of 10, all the papers with less than 10 citations are completely neglected. We are dividing the number of citations equally among different all co-authors. So, if there are two papers

with equal number of citations and one paper is written by one author and second is written by two authors, the author of the first paper will be more highly ranked as compared to the other two authors. Good paper will lead to increase in ranking more easily, unlike to h-index in which one paper cannot affect the value. The disadvantage is that it does not give weightage to the number of papers the author has published. For example, if there are two authors both with metric score of 50, but one has published ten papers and other has published just one, still both of them will be ranked equally, which should not be the case ideally. One excellent paper can cause a huge jump in the ranking. There should be a way to normalize it.

B. Clustering of authors

Out task identify top 100 researchers from different disciplines. For this, we need to first cluster the authors based on their areas of research and interests. We identified Aminer dataset which gave us the areas of research for each author. For example, as per the dataset, **Andrew Ng**, has *"Character Recognition, Text Detection, Unsupervised Feature Learning Reading text, Machine Learning, Scene Images, Better feature representation"* as the areas of research.

Using above keywords, we have developed a clustering algorithm which will divide the authors into a group of 10 clusters where each cluster will contain the authors based on similar areas of research.

We also implemented *k Nearest Neighbours (kNN)* algorithm which uses TF-IDF, a numerical statistic, to cluster the authors, but it failed as the algorithm separates each keyword and then apply the algorithms. For example, if the area of research is "Machine Learning", kNN will divide it into "Machine" and "Learning" and then apply the algorithm, but the word "Learning" alone does not make much sense, so the algorithm gave poor results after clustering. This is the reason why we developed and implemented our own clustering algorithm. It takes the whole keyword i.e. "Machine Learning" and then try to find the nearest cluster to that domain. The pseudo code can be found in the algorithm-1.

C. Clustering of papers

For clustering papers, we have followed a similar approach. But, one thing which is different is that we have also used the clusters of the authors that we derived in the previous article to derive the clusters of the papers, so that the clusters of authors and papers remain the same. The pseudo code can be found in the algorithm-2

Using the above clustering algorithms, we identified ten major areas/domains of research. In the next section, we will explain how we applied the Page Rank on the clusters we derived. Below are those ten disciplines:

D. Pagerank

The pagerank implementation is shown in the figure3. Here ϵ is the damping factor. We have normalized the pagerank output by dividing it by ϵ and multiplying it by $|V|$. The main reason for normalizing the score is that normal pagerank is not directly comparable across different graphs. This normalization eliminated the dependence on the size of graph. [9]

Algorithm 1 Clustering Authors Based on Domains

$clustersMap \leftarrow$ an empty map with key as cluster id and value as the keywords in that cluster
for *author* $i \in 1, 2, , n$ *in the authors* **do**
  $keywords \leftarrow$ get list of domains for the $author$
  $maxMatchingCount \leftarrow 0$
  $maxMatchingLength \leftarrow 0$

  for *cluster* $j \in 1, 2, , m$ *in the* $clusterMap$ **do**
    $matchingLength \leftarrow$ number of $keywords$ matching in the cluster;
    if $matchingLength > maxLen$ **then**
      $maxMatchingLength \leftarrow matchingLength$
      $maxMatchingCount \leftarrow 1$
    end
    else if $matchingLength == maxLen$ **then**
      $maxMatchingCount \leftarrow$ increment by 1
    end
  end
  if $maxMatchingCount == 0$ **then**
    create a new cluster and add in $clusterMap$
  end
  else if $maxMatchingCount == 1$ **then**
    add $author$ to that cluster
  end
  else
    $minSizeCluster \leftarrow$ the cluster with minimum size among all the matching clusters add $author$ to the $minSizeCluster$
  end
end
return $clusterMap$

E. P^2 score calculation for authors

We have named our proposed score metric p^2, as it is a function of 2 different pageranks. We have applied pagerank to following two graphs:

- **Paper citation network**: The paper citation network contains the information about which paper is to related to which other papers. In the citation network a node's incoming edges are the other papers that has cited this paper and it outgoing edges are the references of this paper. The pagerank value of each node(paper) shows its relative importance compared to other nodes(papers). This value can be use to rank the authors as there is direct relation between the importance of the paper and author of that paper. We are doing the sum of all pagerank scores of the papers that have been written by an author as using it as a score for that author.
- **Co-author network**: This network contains the information about an authors co-authors. Every node in the graph is an author and it is connected with its co-author. One can see that an author, working with an author who is distinguished in his field, also should get more importance compared to the author who is working with the similar ranked authors. We are using the pagerank value of a node from this network directly as measure of

Algorithm 2 Clustering Papers Based on Domains

$authorsClusterMap \leftarrow$ the map containing authors with their keywords and clusters
$papersClusterMap \leftarrow$ an empty map to store the clusterID for each paper **for** *paper* $i \in 1, 2, , n$ *in the papers* **do**
  $clusterFrequency \leftarrow$ empty array to store count of each occurance of cluster
  $authors \leftarrow$ all the authors who contributed to this paper
  for *author* $j \in 1, 2, , m$ *in the* $authors$ **do**
    $clusterFrequency[clusterID[author] \leftarrow$ the cluster which occurs for majority of the authors of this paper
  end
  $maxClusterID \leftarrow 0$
  $maxClusterCount \leftarrow 0$
  for *index* $j \in 1, 2, , m$ *in the* $len(clusterFrequency)$ **do**
    if $maxClusterCount < clusterFrequency[j]$ **then**
      $maxClusterCount \leftarrow$ clusterFrequency[j]
      $maxClusterID \leftarrow$ index+1
    end
  end
  $papersClusterMap[paper] \leftarrow maxClusterID$
end
return $clusterMap$

ClusterID	Domain of Research
1	Data Analysis, Information Retrieval
2	Network Control System, Communication Network, Network Performance
3	Image Processing, Video Compression, Binary Image Segmentation
4	Software Engineering, Product Design, Engineering and Development
5	System Performance, Caching and Parallel Computations
6	Graph and Tree Algorithms, Polynomial Functions and Complexities
7	Quantam Computing, Quantam Systems
8	Search and Optimzation Algorithms, Scheduling Problems
9	Power Circuits, FPGA Performance, Hardware Design
10	Neural Networks, Linear and NonLinear Functions

TABLE I: Main keywords covered by the each domain

the importance of that author.

Finally, we calculate the p^2 score. We have used a linear summation of the above two pageranks values. The equation to calculate p_2 is as follows:

$$P^2(author) = (0.25) * P_{coauthor\_network}(author) \; + \; (0.75) * \sum^{Publications} P_{citation\_network}(publication) \tag{2}$$

Algorithm 3 Pagerank

$G \leftarrow$ a network
$P_0 \leftarrow \epsilon/|V|$, where ϵ is damping factor and $|V|$ is total number of nodes in the graph
$k = 0$
while *True* **do**
  $P_{k+1} \leftarrow \epsilon/|V| + (1-\epsilon)A^T P_k$
  $k = k + 1$
  $until P_{k+1} - p_k < \delta$
end
return $P_{k+1}|V|/\epsilon$

A higher weightage is given to pagerank of citation network because it is a more direct measure the authors' importance.

F. Papers Reach Function

We have ranked papers as per their reach in both local and globalized domains. For each of these, we have used different functions to calculate the metric which is explained below:

1) Local Reach

As explained in section 4.3, we divided the papers in a group of clusters/domains which is based on a certain area of research. For each cluster, we then run page rank algorithm which will give us the *reach* for each paper. Since we have run this page rank for each cluster, we will get the result which is localized for the particular cluster. Below is the equation to get the metric:

$$LR(paper) = \sum^{Publications} P_{citation\_network}(publication) \tag{3}$$

2) Global Reach

: This was one of the most challenging parts of the project. To figure out the impact of the paper globally, we implemented an algorithm using cluster of each author which contributed to the paper and the number of citations of the paper from the papers which belong to other domains. This is because if the authors which contributed to the paper belong to different domains and if papers from other domains have cited this paper, this certainly means that the paper is more *Globally Reachable*. Algorithm to calculate metric found in algorithm-4.

G. Future Rank [9]

To identify the papers that should become popular we are using the FutureRank algorithm given by the Hassan Sayyadi and Lise Getoor [9]. Here we are using two networks, the first network only contains the paper nodes and citation edges between them(citation network), so pagerank can be used here for authority transfer from articles to their references. The second network is the network of papers and authors but only contains authorship edges. It can be mapped onto a Hyperlink-Induced Topic Search(HITS) network where articles are authorities and authors are hubs, and the network can simulate

Algorithm 4 Global Reach of a Paper(P, paper_network, author_network)

$total\_clusters\_count \leftarrow 10$ $paper\_network \leftarrow$ citation network of papers
$author\_network \leftarrow$ author-author network
$papers\_cluster\_set \leftarrow$ an empty set to store the unique clusters
$combined\_paper\_page\_rank \leftarrow 0$, this is the combined page rank of all the papers which are citing paper P
for *paper* $i \in 1, 2, , n$ *in the papers* **do**
  if *paper is citing* P **then**
    add $clusterID$ of $paper$ to the $papers\_cluster\_set$
    $combined\_paper\_page\_rank$ += page rank of $paper$
  end
end
$authors\_cluster\_set \leftarrow$ an empty set to store the unique clusters
$combined\_author\_page\_rank \leftarrow 0$, this is the combined page rank of all the papers which are citing paper P
for *author* $i \in 1, 2, , m$ *in the authors* **do**
  add $clusterID$ of $author$ to the $authors\_cluster\_set$
  $combined\_author\_page\_rank$ += page rank of $authors$
end
$global\_reach = (0.8) * combined\_paper\_page\_rank * length(papers\_cluster\_set) + (0.2) * combined\_author\_page\_rank * length(authors\_cluster\_set)$
return $global\_reach$

the mutual reinforcement between articles and authors using a HITS-style propagation algorithm.

If our graph has total P papers and A authors then matrix M^C is $|P|\times|P|$ citation matrix. $M^C_{i,j}$ is 1 if p_i cites p_j, otherwise 0. Further, we have added the edges between dangling nodes to every other node. M^A is a $|P|\times|A|$ authorship matrix. $M^A_{i,j}$ is 1 if a_i is author of paper p_j, otherwise 0.

To give ranking to the papers and authors we will iterate one step of pagerank and one step of HITS algorithm and combine their values. We have repeated this step until convergence. R^P denotes paper rank and R^A denotes author rank and the calculation of these ranks are as follows:

$$R^A = M^A * R^P \tag{4}$$

$$R^P = \alpha * M^C * R^C + \beta * (M^A)^T * R^A + \gamma * R^{Time} + (1-\alpha-\beta-\gamma) * (1/|V|) \tag{5}$$

In other words, R^A collects its authority score from all his publications. In the calculation of R^P $M^C * R^C$ is pagerank score of citation network, $(M^A)^T * R^A$ is the authority score in the authorship network and R^{Time} is a pre-computed vector based on the query time T_{query} and publication time T_i for the paper p_i. the equation to calculate the R^{Time} is as follows:

$$R^{Time}_i = e^{-\rho*(T_{query}-T_i)} \tag{6}$$

One can easily see that the R^{Time} follows the power law distribution and thus favors recently published papers. The initial value of R^P_i is $\frac{1}{|P|}$ and R^A_i is $\frac{1}{|P|}$. This initialization keeps the sum of the paper ranks equal to 1, as well as the sum of author ranks and this will hold after each iteration too, since the sum of weights $\alpha+\beta+\gamma+(1-\alpha-\beta-\gamma)=1$.

IV RESULTS

A. Top 100 authors derived from Baseline Model

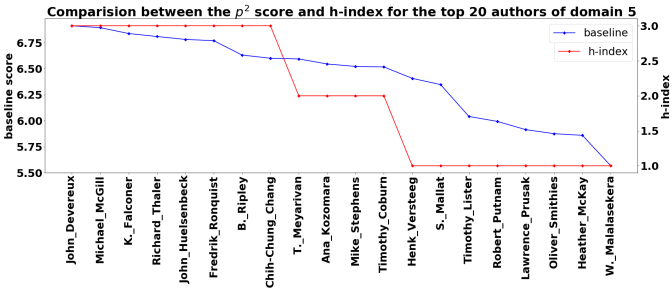

Fig. 1: Comparison between baseline model and h-index

The figure-1 shows the performance of our baseline metric vs the h-index. As seen from the graph, our baseline metric follows the h-index. This index performs even better than h index in some of the cases. For example, authors like Rama Chellappa and Deborah Entrin have same h-index but our baseline index ranks Rama Chellappa above Deborah Entrin.

B. Top 100 authors of each domain

In this report, we have mentioned the name of the top 5 authors from five domains. These can be found in the tables II, III, IV, V and VI. To view top 100 authors for all the domains, you can refer to Appendix.

Author Name	h-index	P^2 score
Yang Liu	67	6.294
Lei Zhang	107	5.635
Li Zhang	72	4.632
Yu Zhang	50	4.475
Jian Zhang	62	4.321

TABLE II: Top 5 Authors in "Data Analysis, Information Retrieval" Domain

Author Name	h-index	P^2 score
Wei Wang	104	3.586
Wei Zhang	83	2.984
Wei Li	77	2.543
Jun Wang	92	2.235
Xin Wang	50	2.217

TABLE III: Top 5 authors in "Network Control System, Communication Network, Network Performance" Domain

Author Name	h-index	P^2 score
Hong Zhang	50	5.004
Jie Yang	70	4.788
Fan Zhang	50	4.274
Wei Xu	60	4.01
Ivanoe Falco	13	3.937

TABLE IV: Top 5 authors in "Image Processing, Video Compression, Binary Image Segmentation" Domain

Author Name	h-index	P^2 score
Yi_Li	50	3.565
Long_Chen	50	3.297
Hai_Jin	50	2.702
Mohammad_Khan	50	2.484
Magdy_Bayoumi	50	2.32

TABLE V: Top 5 authors in "System Performance, Caching and Parallel Computations" Domain

Author Name	h-index	P^2 score
Tao Jiang	63	3.517
Eun Kim	50	3.136
Kenichi Kawarabayashi	50	2.553
Xiao Zhang	50	2.233
Pankaj Agarwal	50	2.191

TABLE VI: Graph and Tree Algorithms, Polynomial Functions and Complexities

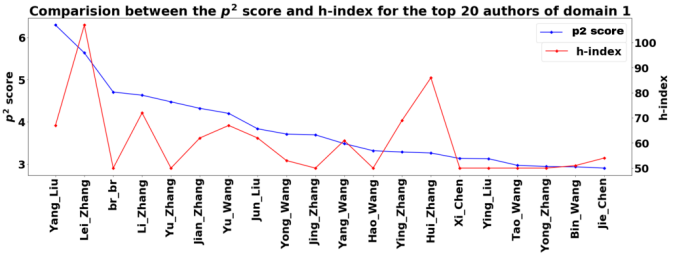

Fig. 2: Comparison between p^2 and h-index of top 20 authors of domain "Data Analysis, Information Retrieval".

We can see in the above graph that P^2 scores are following a similar trend as h-index. The author with higher h-index are also getting higher P^2 score. The graph for h-index is very kinky, reason for that could be that h-index can only be integer. We can see in other graphs as well that a similar trend is being followed in all the domains.

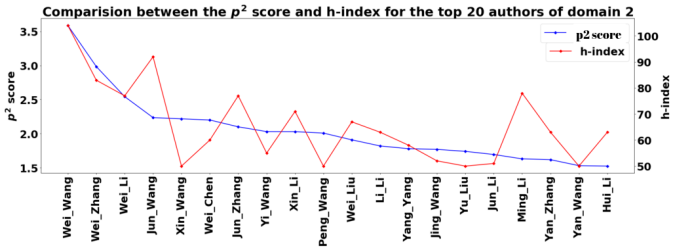

Fig. 3: Comparison between p^2 and h-index of top 20 authors in domain "Network Control System, Communication Network, Network Performance"

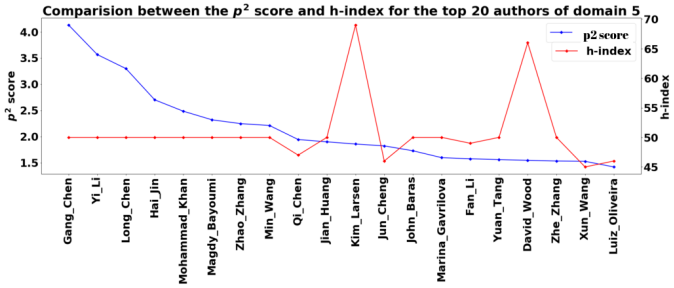

Fig. 4: Comparison between p^2 and h-index of top 20 authors of domain "System Performance, Caching and Parallel Computations"

The top 100 authors of each domain can be found in appendix section. Out of which we have plotted the graphs of top 20 authors of 3 randomly chosen domains and compared their p^2 score and h-index. See figure-2, 3, 4 for details. One can see that there some similarity between p^2 and h-index. One of the major problem with the h-index is that it cannot take float values, so it is hard to rank the authors that has same h-indices. As the p^2 score can take float values as well we can compare authors at fine grain compared to h-index.

C. Comparison between $\mathbf{P^2}$ and h-index

We try to calculate similarity between p^2 score and h-index using Pearson and Spearman coefficient and result are as follows:

- $r_{Pearson\_coefficient}(P^2, h-index) = 0.4142$
- $r_{Spearman\_coefficient}(P^2, h-index) = 0.4484$

We can see that both the coefficients suggest that the h-index and the P^2 are correlated. This was expected as both of them use the citations in some way or the other.

1) Anomaly in H-Index

In the table VII, we have shown some authors where the h-index fails to index them correctly. One can observe that our index p^2 has given more score to rest of the authors compared to first author. The first author has written 98 papers where as rest of authors have written more than 150 papers and have total citation more than 1000. Author Peter Loos has scored less compared to Hesham Ali and Imre Rudas because the paper pagerank score is far less than other two authors.

Author Name	Giovanni Acampora	Peter Loos	Hesham Ali	Imre Rudas
Paper count	98	213	153	165
Number of citations	943	1390	1003	1098
H-index	43	40	30	39
Normalized paper page-rank	4.17	2.35	3.99	2.8
Normalized co-author page-rank	4.00	11.6	10.93	13.5
Combined normalized page-rank score	4.12	4.68	5.61	5.50

TABLE VII: Comparison between h-index and our p^2 score

2) H-Index biased towards large clusters

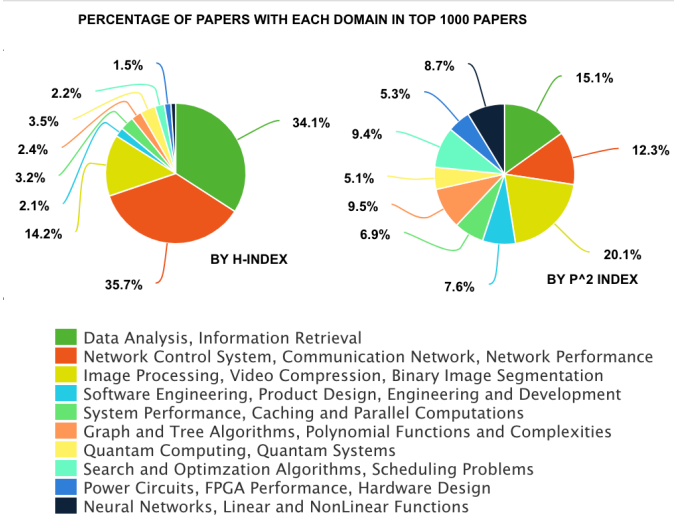

Fig. 5: Comparison between number of authors in the top 1000 by the p^2 score and h-index domain wise

We can see in the above graph that if we get top 1000 authors based on h-index, most of them are from larger clusters. This seems to be biased because of the cluster size. On the other hand, P^2 is much less biased. The same can be seen in the figure-5.

D. Paper Reach

1) Local Reach

: This tells us which paper is more important within each cluster, i.e. the paper which is cited by many papers within this domain. You can find the top 2 papers with highest reach of 5 clusters in table-VIII.

2) Global Reach

Below are the top 10 papers with maximum global reach i.e. the papers with highest interdisciplinary impact.

- *Perceived usefulness, perceived ease of use, and user acceptance of information technology*
- *Statistical mechanics of complex network*
- *Speeded-Up Robust Features (SURF)*
- *A method for registration of 3-D shapes*
- *Authoritative sources in a hyper-linked environment*

Paper Title	Pagerank	Cluster ID
ArnetMiner: extraction and mining of academic social networks	7.888	1
The International Exascale Software Project roadmap	5.296	1
Deep speech 2: end-to-end speech recognition in English and mandarin	6.193	2
Rethinking energy efficiency models of cellular networks with embodied energy	5.928	2
Fuzzy reasoning spiking neural P system for fault diagnosis	5.129	3
Joint access and power control in cognitive femtocell networks	4.427	3
Software Engineering for Self-Adaptive Systems: A Research Roadmap	4.099	5
Automated support for classifying software failure reports	4.071	5

TABLE VIII: We have shown the top 2 papers of each 5 clusters with the highest reach in their local domain

- *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data*
- *An introduction to variable and feature selection*
- *Statecharts: A visual formalism for complex systems*
- *Case-based reasoning: foundational issues, methodological variations, and system approaches*
- *A taxonomy and evaluation of dense two-frame stereo correspondence algorithms*

E. Future Rank

Below are the top 10 papers from the results of our algorithm to calculate which papers should become popular.

- *Construction and Analysis of Weighted Brain Networks from SICE for the Study of Alzheimer's Disease*
- *Deep speech 2: end-to-end speech recognition in English and mandarin*
- *Learning to Play in a Day: Faster Deep Reinforcement Learning by Optimality Tightening*
- *Three-dimensional texture features from intensity and high-order derivative maps for the discrimination between bladder tumors and wall tissues via MRI*
- *Effects of process and outcome controls on business process outsourcing performance: Moderating roles of vendor and client capability risks*
- *Modeling Active Virtual Machines on IaaS Clouds Using an M/G/m/m+K Queue*
- *A Framework for Practical Dynamic Software Updating*
- *Person re-identification by multiple instance metric learning with impostor rejection*

- *A method considering and adjusting individual consistency and group consensus for group decision making with incomplete linguistic preference relations*

V CONCLUSION

In this report, we have successfully developed a metric to counter the failures of the author's h-index. We gave few examples where we can clearly see h-index failing and also showed why comparing authors of different fields with h-index is a bad idea. We applied page rank for different clusters separately, normalized the page rank value based on the size of the clusters and we were able to successfully compare the authors of different areas. Also, we have compared the papers in both "intra-domain" and "inter-domain" and rank the papers based on their impact. At the end, we also calculated which papers should become more popular in the future.

REFERENCES

[1] Eugene Garfield. ""Science Citation Index"-A New Dimension in Indexing"". In: *Science* 144.3619 (1964), pp. 649–654.

[2] Jorge E Hirsch. "An index to quantify an individual's scientific research output". In: *Proceedings of the National academy of Sciences* 102.46 (2005), pp. 16569–16572.

[3] Václav Vavryčuk. "Fair ranking of researchers and research teams". In: *PloS one* 13.4 (2018), e0195509.

[4] Leo Egghe. "An improvement of the h-index: The g-index". In: ISSI. 2006.

[5] Leo Egghe. "Theory and practise of the g-index". In: *Scientometrics* 69.1 (2006), pp. 131–152.

[6] Bihui Jin, Liming Liang, Ronald Rousseau, and Leo Egghe. "The R-and AR-indices: Complementing the h-index". In: *Chinese science bulletin* 52.6 (2007), pp. 855–863.

[7] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong 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.

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

[9] Klaus Berberich, Srikanta Bedathur, Gerhard Weikum, and Michalis Vazirgiannis. "Comparing apples and oranges: normalized pagerank for evolving graphs". In: *Proceedings of the 16th international conference on World Wide Web*. ACM. 2007, pp. 1145–1146.

[10] *KDD cup 2003*. http://www.cs.cornell.edu/projects/kddcup/datasets.html. 2003.

[11] https://dev.elsevier.com/documentation/ScopusSearchAPI.wadl#d1e33.

[12] https://pypi.org/project/scholarly/.

[13] https://arxiv.org/help/bulk_data.