1-2017

The habits of highly effective researchers: An empirical study

Subhajit DATTA
Singapore Management University, subhajitd@smu.edu.sg

Partha BASUCHOWDHURI
Heritage Institute of Technology

Surajit ACHARYA
Heritage Institute of Technology

Subhashis MAJUMDER
Heritage Institute of Technology

IEEE TRANSACTIONS ON BIG DATA, VOL. 3, NO. 1, JANUARY-MARCH 2017
DOI: 10.1109/TBDATA.2016.2611668

The Habits of Highly Effective Researchers: An Empirical Study

Subhajit Datta, Partha Basuchowdhuri, Surajit Acharya, and Subhashis Majumder

Abstract—Interest in the habits of influential individuals cuts across domains. As researchers, we are intrigued why few attain significant eminence in their fields, whereas many operate in obscurity. An empirical examination of this question has been made possible by the recent availability of large scale publication data. In this paper, we use information from the *AMiner Paper Citation and Author Collaboration Networks* to discern factors that relate to the impact of influential researchers across five domains in the computing discipline. We propose and apply a novel algorithm to identify influential vertices in co-authorship networks built from total corpora of 1,00,000+ papers and 72,000+ authors over a span of more than 50 years. The results from our study indicate that the impact of these influential researchers relate to a variety of factors. Surprisingly, we find evidence across the domains that higher impact is associated with lower levels of collaboration, and authority.

Index Terms—Big data, social network analysis, graph algorithms, dominating sets, software engineering, networking, operating systems, databases, artificial intelligence

1 Introduction

In the world of scientific publications, influential authors have an inordinately large impact on the direction of a discipline [1]. Often, the accretion of ideas around these individuals define the contours of the discipline, as well as its future path. With increasing focus on the *science of science* [2], habits that relate to an individual becoming influential in a particular area is of interest to researchers and practitioners. In recent times, public domain repositories such as Aminer[1] facilitate empirical examination of the habits of highly influential researchers. In this paper, we report results from a large scale study across five computing domains to uncover statistically significant evidence on the factors relating to the influence of researchers in the community.

In our realm of interest, questions such as what is influence, and how to identify influential researchers are far from settled. Influential researchers definitely publish and get cited more than other, non-influential ones. However, as is widely recognized, measuring influence solely by a single metric is an incomplete, and at times a misleading approach [3]. In a research ecosystem, influence is closely associated with the spread of ideas and how individuals participate in that process. Such spread is affected by-and in turn affects-the so-called "network effects" [4] of collaborative research. In this paper, we first present an algorithm for selecting the influential set of researchers from the co-authorship network of research publications. We next run the algorithm on our data-sets from the domains of software engineering (SE), operating system (OS), databases (DB), artificial intelligence (AI), and networking (NW) to determine the influential set of authors of our interest[2] in each domain. Various characteristics of the members of these sets are then statistically analyzed to determine the habits of influential researchers that relate to their impact. The study reported in this paper addresses the following research question:

In addition to publication and citation count, what other factors relate to the research impact of highly influential researchers?

1. https://aminer.org/

- *S. Datta is with Singapore University of Technology and Design, Singapore 487372. E-mail: subhajit.datta@gmail.com.*
- *P. Basuchowdhuri, S. Acharya, and S. Majumder are with the Heritage Institute of Technology, Kolkata, West Bengal 700107, India. E-mail: {parthabasu.chowdhuri, surajit.acharya, subhashis.majumder}@heritageit.edu.*

2 Motivation and Outline of our Approach

Measuring the impact of individual researchers remains a key component of scientific ecosystems. In early 20th century, Cattell first proposed the notion of systematically ranking scientists by their "performance" and highlighted how such ranking could be useful [5]. In the mid 1960s, Sher and Garfield's pioneering work on indexing scientific literature using punch cards led to lasting insights and the definition of one of the most widely used citation-based metric-the "impact factor" [6]. While the impact factor remains popular even today, in bibliometric circles it is considered to be "mortal sin" to use a journal's impact factor to measure the research performance of an individual researcher [7]. Results from principal component analysis of 39 existing and proposed impact measures indicate that scientific impact is essentially multi-dimensional and a single indicator can not measure it

2. In the remainder of this paper, "author" and "researcher" are used interchangeably; and "vertex" and "node" are used interchangeably in the network context.

TABLE 1
Existing Ways of Measuring Impact Vis-a-Vis Our Approach

Metric/Model	Definition/Context	Remarks
Number of citations	This is computed by counting the number of times a researcher's papers are cited by others. The metric reflects the extent to which others recognize a researcher's past published work. As this is a raw count, it is difficult to use this metric to compare researchers in different fields or at different stages in their scientific career [7].	We consider number of citations as a control variable in our models (Section 7), along with the number of publications, to account for its effects across different domains and career stages.
h-index	This metric has gained wide popularity for measuring productivity and impact after being proposed in 2005 [12]. A researcher has a h-index of n, if (s)he has n publications, each of which has been cited at least n times by others. Range of h-index values for effective researchers vary across different fields, and the h-index value for a particular researcher can not decline with retirement or reduction in research output. Many variants of h-index are also used, such as the *contemporary h-index* which gives more weight to recent articles, and to the *g-index*, which gives more weight to highly cited articles [7]. It has been suggested that h-index is also useful for predicting future performance of researchers [13].	The h-index has been used as the dependent variable in our models (Section 7).
Impact factor	This is defined as the frequency with which an average article in a journal gets cited [7]. For example, the impact factor of a journal in 2016 would be the total number of citations its articles received in 2014 and 2015 divided by the number of "citable" items published in the journal during these two years.	As discussed in Section 2, impact factor for a journal is unsuitable for calculating the impact factor of an individual researcher. Thus we do not consider this metric in our models.
Weighted citations	Similar to Google's PageRank algorithm, a citation from a popular article or researcher is weighted more heavily in calculating the weighted citations metric [7]. Although availability of large scale bibliometric data online makes it easy to compute this metric, there is no standardization yet in place for applying weighted citation to measure individual impact.	Due to its lack of applicability at the individual level, we have not considered this metric in our models.
Online accesses	This metric is calculated as the number of times a research paper is accessed or downloaded online [7]. Although this metric is more up-to-date than citation count, global standards on reporting are not yet in place [14]. Additionally, online attention or "eyeballs" may not be a reliable proxy for scientific interest in a researcher's work.	Due to lack of standardization, we have not used this metric in our models.
Betweenness centrality	This is a measure of how a vertex is placed on all pathways connecting other vertices in a network [7]. This is one among several metrics that quantify how researchers are interconnected. However, it reduces a network to a single number, thereby sacrificing much of the richness of information of the network structure.	We have included betweenness centrality as an independent variable in our models (Section 7).

adequately [8]. Qualms have also been raised about pitfalls of quantitative measurement of research impact [9], and how metrics can send mixed messages, often influenced by the context [10].

In this paper, we posit that while it is important to measure scientific impact at the individual level, there is a more fundamental dynamic that needs to be investigated: *what are the factors that influence the success of effective researchers?* As Kuhn pointed out, effective researchers build a community of influence around them [1], something of an "invisible college" in de Solla Price's words [11]. This leads us to believe that the first step in investigating factors behind the success of researchers, is to identify influential researchers in a networked research ecosystem. Having identified such researchers, statistical analysis can lead to factors contributing to their success. Accordingly, in this paper we first present a general algorithm for identifying influential vertices in a network (Section 5). In the context of our study, this algorithm can run on a co-authorship network of a research domain to identify a dominating set of researchers, who are best positioned to influence percolation of their ideas in that community. We then apply the algorithm on co-authorship networks across five research domains (Section 6), followed by developing statistical models (Section 7) and deriving insights from them (Section 8). The paper ends with a discussion of the utility of the results (Section 9), threats to validity (Section 10), and conclusions (Section 11).

Table 1 positions our work as it complements existing ways of measuring scientific impact. We have selected the existing measures from a "Field guide to metrics" as identified in [7], published in Volume 465, June 17, 2010 of *Nature*, which was focused on assessing measurements of scientific impact.

From the above discussion and the summary presented in Table 1, it is apparent that understanding factors that

influence researcher impact is fraught with challenges. To address these challenges, it is recommended that the entire range of a researcher's work be considered [15], [16]. Accordingly, we consider a range of factors in this work (as described in Section 7) to investigate influences on effective researchers.

In the next two sections we highlight our research contributions and give an overview of related work, respectively.

3 Research Contributions

The major research contributions of this study are:

1) We present a novel algorithm for determining the influential set of vertices in a network and illustrate its use on co-authorship networks. In addition to the use illustrated in this paper, the algorithm can be applied in diverse contexts such as selecting individuals who can spread a product's reputation for a viral marketing campaign, to ensure the campaign achieves high penetration within a short period of time.
2) Using large-scale data across different domains, we analyze factors that relate to the success of influential researchers. Our results uncover many of the dynamics that underpin the world of scientific publications, leading to a more complete view of what it takes to be effective in research. The insights from this study can inform individual as well as collective decision-making in the pursuit research impact.

4 Related Work

In this section, we give a brief overview of existing work related to selecting influential nodes from a network, and bibliometric analysis of scientific publications.

4.1 Selecting Influential Nodes from a Network

Selecting influential vertices from a social network is a well-studied problem. Researchers working in social networks have used different measures to identify a node's level of influence. Some of them follow traditional graph mining techniques [17], [18]. There could be local reachability-based metrics like degree centrality [19] that measure how important a node is, within a part of the network. An alternative representation of the degree centrality, otherwise known as Bonacich's power centrality [20] argues that depending on the situation, a node could be more important to the network in terms of reachability when its neighbors are not important. This directly contradicts the idea of PageRank [21], which also aims to find influential nodes in a network but with the idea that a node is important if its neighbors are important. Bonacich's power centrality nicely accommodates both the contradictory ideas using a parameter β, which is tuned depending on the problem or the dataset. There are other shortest-path distance-based measures like closeness [22], which measures how quickly a piece of information can reach the rest of the nodes in the network. Understandably, such calculations are dependent on time-consuming single-source shortest-path algorithms. Similarly, another measure named betweenness [23] measures occurrence of a node within the shortest paths between all-pair of nodes other than the node itself. Higher betweenness of a node signifies more brokerage value. A node with higher betweenness value may be involved in the passage of the information frequently but it may not be central or influential in terms of reachability. Table 1 summarizes the relevance of betweenness for our study, among other factors.

Another way of finding influential nodes in terms of the spread of information may be by finding the minimum dominating set [24], [25], [26], [27], [28]. This is a well-known NP-hard problem. Therefore, finding optimal solution may not be possible. But we may still try to find a sub-optimal solution that gives us a reasonably small group of nodes that would dominate the rest of the nodes in the network. We call such group of nodes, which can dominate the rest of the nodes, as *seed* set. A more generic version of the problem is the k-hop dominating set problem [29], [30]. Increase in k leads to increased number of nodes dominated by each node in the seed set and therefore, with increase in k, a reduction in the size of the seed set is evident. Identifying a minimum dominating set addresses the one-hop version of the problem. There are known greedy approaches for finding a minimum dominating set [29], [31]. Also, sometimes approximation algorithms have been used to find minimum dominating sets in different contexts [32]. With this background, Section 5 presents our algorithm for identifying influential authors.

4.2 Bibliometric Analysis of Scientific Publications

[Evolution of Scientific Collaboration]. Barabasi et al. report results from a pioneering study of scientific collaboration in [33]. Co-authorship networks in mathematics and neuroscience in the period 1991-1998 are explored, and the authors conclude these networks are scale-free and network evolution is governed by preferential attachment. The authors also propose a model to explain the networks' evolution. However, in this study, collaboration is examined for a limited time-window, some of the observations-such as dramatic growth of the largest connected cluster-are ascribed to the "missing past" problem, that is, the period of time since the beginning of organized publication in the discipline outside the time-period of the study. Newman has examined the structure of scientific collaboration in detail; he shows that such collaboration networks form small-worlds where pairs of randomly selected scientists are typically short distances away from one another and the networks are significantly clustered [34]. The author takes forward his exploration in subsequent papers, where properties of co-authorship networks are studied, along with the existence and size of a giant component, and other non-local characteristics [35], [36]. These papers illustrate how scientific collaboration in different disciplines manifest subtly different patterns. Interestingly, our results from studying five different domains in the computing disciplines reveal certain commonalities in the characteristics of effective researchers across domains (Section 8). The innate differences between social networks and other types technological or biological networks are studies by Newman and Park in [37].

[Understanding the Science of Science]. The impact of co-authorship teams are studied by Boerner et al. using a set of 614 articles by 1,036 authors between 1974 and 2004 [2]. The

authors find a trend towards deepening global collaboration in the pursuit of scientific knowledge. While these results may be valid at the global level, for individual researchers, we find evidence of less collaboration to be associated with higher impact (Section 8). Bettencourt et al. study publication data from multiple research areas and conclude that, while each field develops differently over time, population contagion models can explain their development [38]. The dynamics of scientific disciplines is studied by Herrera et al. [39]. The authors construct an idea network of American Physical Society's Physics and Astronomy Classification Scheme (PACS) numbers as nodes representing scientific concepts and use a community finding algorithm to detect how these fields have evolved between 1985-2006. The most influential documents in a corpus are identified by a dynamic topic model proposed by Gerrish and Blei; they validate their model on three corpora-selected publications from the Association for Computational Linguistics (ACL) anthology, the Proceedings of the National Academy of Sciences (PNAS) and the journal *Nature* [40].

[Research Collaboration in Computing]. Evolution of research collaboration networks based on co-authorship in computer science in the period 1980 to 2005 have been studied by Huang et al. [41]. They examine six subcategories within computer science and conclude that the database community is the best connected. Bird et al. define 14 sub-areas within computer science and use topological measures to examine how collaboration patterns vary across areas. Hassan and Holt study co-authorship networks from the proceedings of the Working Conference on Reverse Engineering (WCRE)-for the period 1993-2002 and conclude that these have properties of small-world networks [42]. Glass, Vessey, and Ramesh have studied 369 papers in six software engineering publication venues and to infer that software engineering research is "... diverse regarding topic, narrow regarding research approach and method, inwardly-focused regarding reference discipline, and technically focused ... regarding level of analysis" [43]. These authors have also compared "three major subdivisions of the computing realm"-computer science, software engineering, and information systems-and discover that each field has preferred research approach and methods, which is not necessarily "respected" by the other fields [44].

4.3 Big Data and Its Applications

Increasingly, big data is finding applications beyond its conventional areas of influence. General directions of big data analysis are discussed at length in [45], [46]. Hashem et al. examine the relevance of big data on cloud computing and the relationship between big data and cloud computing, big data storage systems, and Hadoop [47]. In a subsequent paper, Hashem et al., investigate the role of big data in smart cities and propose a future business model along with identifying business and technological research challenges [48]. Yaqoob et al. survey the application of information fusion to social big data and highlight its benefits and challenges [49].

This background of related work will help position our algorithm and its validation.

5 Identifying Influential Authors

We now present our algorithm for identifying influential authors in a co-authorship network.

5.1 Problem Definition

Given an undirected, unweighted social network $G(V, E)$, assume that some information needs to be spread to all the nodes across the network. By k-hop domination, we mean any node u that is chosen as a seed node is capable of spreading information to all those nodes that can be reached from u within k hops.

We now define below a few relevant terms that are needed to formally introduce the problem that we are addressing here.

Definition 1 (k-hop Neighborhood of a vertex). *In an undirected graph G, a vertex v is said to be present in the k-hop neighborhood of a vertex s, if there exists a simple path from s to v of length $\leq k$.*

Definition 2 (k-hop Dominating Set (kHDS)). *Given an undirected graph $G = (V, E)$, a k-hop dominating set is a subset $S \subseteq V$ of its vertex set such that any node $v \in V \setminus S$ is in the k-hop neighborhood of s, denoted by $N_k(s)$, for some $s \in S$.*

If S is a kHDS of a graph G then,

$$V(G) = S \cup (\bigcup_{s \in S} N_k(s)).$$

The set S is often called a seed set. A k-hop dominating set of minimum cardinality is called a k-hop minimum dominating set (kHMDS). Identifying a kHMDS for a general graph is computationally hard [29], however, finding it for a tree can be done in polynomial time [30]. We present below a fast heuristic for finding kHDS of small size for graphs.

5.2 Finding a k-Hop Dominating Set for a Tree

We start with a version of our algorithm that would find the optimal k-hop minimum dominating set from a tree. In this algorithm, a tree G_T and a value for k (number of hops) will be the input and corresponding kHMDS will be the output. If the tree is unrooted, we pick any node r as a root node and re-arrange the tree as a rooted tree by running a breadth-first search from r. All the nodes of the resulting breadth-first tree are assigned levels with r having level 0. Any node with a BFS distance of d from r is assigned a level d. Note that this initial step consisting of identifying a breadth-first tree spanning over all the nodes and then assigning levels remains same even in the case of graphs. We have presented the pseudo-code of this algorithm for tree in Algorithm 1 of the Appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TBDATA.2016.2611668, where we deal with three major concepts-

Upward Traversal from Leaf Nodes. In order to cover all the nodes in the graph we have to cover the leaf nodes also. We start our method by selecting a leaf node with highest level and traversing upwards along the tree edges to select its predecessor at k-hop distance. In Lemma 2, we prove that in case of a tree, to cover a leaf node, picking the kth predecessor as an influential node is the optimal choice.

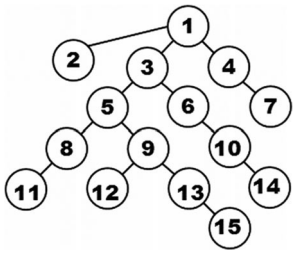

Fig. 1. A test case for running T-kHMDS.

Covering all the Sub-Trees from Selected Influential Node. Once an influential node s has been selected, all the nodes in the sub-tree rooted at s will be covered. Other than the successors of the influential nodes, there could be other nodes, which are within k hops from the influential node. For example, a sibling of an influential node, in case of $k = 2$. We could have covered all such nodes by running a breadth first traversal for k hops using s as root. However, we avoid this direct approach in order to ensure the linearity of our proposed algorithm.

Setting hopCount for Certain Nodes. We introduce a distance counter named $hopCount$ for marking the predecessors of an influential node on the basis of its distance from the influential node s. The primary purpose for introduction of $hopCount$ is to cover those nodes present in the sub-trees rooted at the predecessors of s, which are within the distance k from s. However, these nodes will not be marked covered by s immediately. Instead they will be marked covered later and only once. If we do not do so, some of these nodes may get covered unnecessarily by multiple influential nodes, which in turn may violate the linearity of our algorithm.

For explaining the algorithm, we use colors to understand the state of the nodes. The nodes with *white* color are the nodes, who have not been influenced yet, whereas, the nodes with *grey* color are the ones that have been influenced already. We also use the *black* color to mark some special nodes, which makes our algorithm more efficient. A *black* node indicates that it is influenced and the sub-tree rooted at that node has already been covered and therefore in future, by checking that flag, we can stop redundant traversal of the tree edges further down from that node. We just state in advance that all the nodes in the dominating set S (i.e., the set of influential nodes), will be colored *black*. However, there might be some other nodes also that do not belong to S, but will also be colored *black*.

We illustrate this algorithm with an example. We use the tree in Fig. 1 to explain the different aspects of Algorithm 1 of the online Appendix. Say, in this case, value of k is 3 and node 1 is chosen as the root node to create the initial breadth-first tree. According to our algorithm, we will then start from node 15. From node 15, we traverse upwards for k (i.e., three steps) to reach node 5. Here, five is selected as an influential node. It is then added to S, and it is also marked *black*. With a downward k-hop BFS, node 5 covers nodes 8, 9, 11, 12, 13 and 15. Next task is to set $hopCounts$ for the k-hop predecessors of node 5. Traversing upwards from node 5, we reach node 3 and 1 and set their $hopCount$ values to be 1 and 2, respectively. We also mark them as covered by node 5. All the nodes that are covered in the process are colored *grey*. Note that the upward traversal ends at node 1, without reaching k many predecessors, because node 1 is the root node and no further upward traversal is possible.

Next *white* node with highest depth is 14. So, 14 is chosen as the starting point for the next upward traversal in search for the next influential node. A k-hop upward traversal from node 14, points to node 3 as the next influential node. Before reaching node 3, no other node is encountered that has a $hopCount$ value lesser than 3. So, node 3 being the next influential node, we add it to the set S and also color it *black*, and then run a downward k-hop BFS from it. The BFS covers nodes 6, 10 and 14. It also reaches node 5 only to find that its color is *black*. Therefore, further downward traversal from 5 is abandoned. Had we not colored node 5 *black* earlier, from which we have run a downward BFS already, downward k-hop BFS from node 3 would have redundantly covered nodes 5, 8, 9, 11, 12 and 13, thereby increasing the running time of the algorithm. From node 3, another upward k-hop traversal is run to set the $hopCounts$ for its predecessors. Here, the only predecessor node 1 had a $hopCount$ value of 2 previously (when covered by node 5). Since, node 1 is one hop away from the new influential node 3, $hopCount$ value of node 1 should be updated to 1, thereby increasing the chance of covering more nodes by the new influential node. Next source node for upward traversal to find an influential node would be node 7, as it has the greatest depth among the set of nodes that are still uncovered. By upward traversal, it reaches to node 1 in two hops. Since for node 1, $hopCount$ value is set, we now check whether the number of hops used to reach node 1 and its $hopCount$ value is $\leq k$, and we find that it is 2 + 1 = 3 (which is k in this case). So, we conclude that node 7 can be covered by the node that covers node 1 (i.e., node 3). In the next step we run a downward $(k-1)$-hop BFS from node 1 and also color it *black*. Note that, this is an example of a node that is not influential but is still colored *black*. Also, the nodes 2, 4 and 7 that are covered by this BFS are actually covered by node 3 and not node 1. If instead its $hopCount$ was h, we would have to run a downward $(k-h)$-hop BFS from there. However, there exists a more subtle point here that needs to be mentioned. Since we started the last traversal from node 7, this downward BFS cannot go beyond the depth of 7, and hence, it will be automatically restricted to $(k-1)$ hops, even if we run an uncontrolled BFS downward from node 1. Note that if we had not updated the $hopCount$ value of node 1, from 2 to 1 earlier, node 7 could not have been covered by node 3, and as a result node 1 might have been unnecessarily added to the set of influential nodes. Node 3 being colored *black* already, will restrict the reach of downward traversal further down from itself, thereby avoiding redundant traversal. Also, the fact that we color node 1 *black* would provide similar help, if there were nodes present at a higher level than node 1, from where downward BFS could have reached node 1, and intended to go further down. We next show that essentially the notion of $hopCount$ makes sure that Algorithm 1 of the online Appendix remains linear.

Lemma 1. *Algorithm 1 of the online Appendix runs in $O(|V|)$ time.*

Proof. Step 2 being a BFS on a tree ($|E| = |V| - 1 = O(|V|)$), runs in $O(|V|)$ time. Hence Steps 1 through seven together run in $O(|V|)$ time. Step 8 may run several times, however, the cumulative running time is $O(|V|)$. It is guaranteed that each time the while loop executes at least one new node will be covered.

The upward traversal, described in the for loop starting in Step 10, can traverse each edge of the graph at most once. This is because each of these traversals can start only from a *white* node, and may terminate in a node (either influential or with $hopCount$ set to a suitable value) that then turns *black* and immediately after that all its successors turn *grey*.

Note that we can color a node *grey* while traversing downwards (in either the forall loop starting in Step 15 or the one starting in Step 29) only once and hence all such traversals together take $O(|V|)$ time.

The only block that remains is the for loop starting in Step 37. Note that the upward traversal, that is used to cover the k ancestors of an influential mode, may actually cover the same node more than once for different influential nodes and hence may set the $hopCount$ value of the same node again and again. However, we show below that the number of such operations is bounded linearly. Note that for each node being added to S, at most k other nodes may be assigned covers in this for loop (less than k, if root is reached before k steps end). So, in the worst-case, in totality, the number of such assignments possible is k times $|S|$. However, to add a new node s in S, we start traversal from a white node and must traverse k nodes (that are not in S) to reach s. So when we mark the whole sub-tree rooted at s as covered by s, there must be at least k such nodes (apart from s itself) in it that are not in S. Also note that these nodes were not considered at all when the earlier nodes entered S, because sub-trees rooted at influential nodes that are already in S were made unreachable by coloring the influential nodes *black*. So, we conclude $|S| \leq \lfloor \frac{n}{k+1} \rfloor$ where $n = |V|$ and $n \geq k$ (trivially, $|S| = 1$, for $n < k$). In fact, this is a more general result, which will hold for any graph (not only tree) and gives an upper bound on the size of k-hop influential set.

If number of times cover is assigned is given by A then,

$$\begin{aligned} A &\leq n + k|S|, \\ &\quad \triangleright n \text{ from downward \& } k|S| \text{ from upward traversal} \\ &\leq n + \frac{kn}{k+1} \\ &< 2n = O(|V|). \end{aligned}$$

□

Hence the whole algorithm runs in linear time with respect to the number of vertices.

5.3 Selection of Influential Nodes

The tree algorithm starts by picking the unvisited leaf node u with highest depth, say d, from the rooted breadth first tree. Starting from u, k hops are traversed upwards along the tree-edges and the kth predecessor $\pi_k(u)$ is selected as an influential node.

Lemma 2. *For the tree, there exists a kHMDS that contains $\pi_k(u)$ or in other words the choice of $\pi_k(u)$ as an influential node is optimal.*

Proof. By the definition of k-hop cover, in order to cover the node u, we need some node s to be present in the influential set such that it is within distance k from u. So any node whose level value is less than that of $\pi_k(u)$ will not be able to influence u. Also any node v in the tree, whose only path to u goes through $\pi_k(u)$ cannot also influence u, as its distance from u will be greater than k. Hence s must be either $\pi_k(u)$, or belong to the sub-tree T' rooted at $\pi_k(u)$. However, if we choose an influential node with a level value more (i.e., lower) than that of $\pi_k(u)$, then it might dominate less number of nodes from outside the sub-tree T'. Hence we cannot have a better choice than $\pi_k(u)$ as an influential node. Any other choice may lead to increase in the size of S, depending on the tree structure. □

We extend this algorithm for finding a k-hop minimum dominating set from a graph. In the process, the algorithm goes through a number of changes and tackle a few challenges, which were not present in the tree version of the problem. The primary idea is to extract a spanning tree from the graph using BFS and then identify the influential nodes that cover this extracted tree using Algorithm 1 of the online Appendix. Note that a set of nodes that covers this extracted tree will definitely cover the whole graph, rather such a set may be superfluous. This is because the original graph will have extra edges than its spanning tree, which means the shortest path distance between any two nodes of the graph is actually bounded on the upper side by their distance in the tree. In other words, in the graph the same influential node can actually cover more nodes than in the tree. Now, extraction of the tree from the graph may have some effect on the quality of the output of the algorithm, however it is outside the purview of this paper, and here we mainly concentrate on describing the tree-based heuristic for finding the k-hop dominating set. For tree, we have already shown that it is possible to find an optimal kHMDS in $O(|V|)$ time. Next, we present a heuristic (we call it tree-based heuristic) that extends the tree algorithm to find a k-hop dominating set for any graph. We form this heuristic by minimally exploring the cross-edges, i.e., we minimize the part that contributes to the non-linearity in computation but we still use some of the novel features of the tree algorithm to serve our purpose.

The complete method has been described in Algorithms 2, 3, 4, and 5. Algorithm 2 of the online Appendix, which is very much similar to Algorithm 1, is the top-level algorithm that calls Algorithms 3 and 4. Algorithm 3 takes care of the downward k-hop BFS and may in turn call Algorithm 5, if it encounters any cross-edge while traversing downwards. Algorithms 4 and 5 may call each other, i.e., they are mutually recursive but the total number of such calls at any one instance, i.e., the depth of recursion is obviously bounded by k.

We explain the algorithm with an example as seen in Fig. 1 of the online Appendix. In Fig. 1a of the online Appendix, we first show the breadth first tree of the given graph with node 15 being arbitrarily chosen as its root. Cross-edges have been shown by dashed lines.

In this example, as a first step we start with a leaf node with greatest depth, i.e., node 14. Let us assume, here k equals to 2. So, by starting from node 14, a traversal of two hops upwards along the tree-edges is made to reach to the first influential node, which is node 10. If we chose five as

an influential node or a node further up, it won't have been able to influence 14. So to influence 14, choosing either node 13 or node 10 is essential. However, by choosing node 13, we cannot influence node 1, which is why node 10 is a better choice for being in S. The above example basically illustrates the implication of Lemma 2.

5.4 Covering the Descendants from an Influential Node

Once we select an influential node, the immediate next step is to cover all the descendant nodes that are reachable from the selected influential node and this is done in Step 22 of Algorithm 2 of the online Appendix by making a call to Algorithm 3 of the online Appendix. To keep the process linear in terms of running time, we run a k-hop BFS from the influential node only along its child nodes and descendants but we restrict the traversal along the cross-edges. In our implementation, we use a flag named *covered* that stores which node is covered by which node. Once a node is covered by an influential node, it will not get covered by any other node at a later stage. It should be noted that the predecessors are not covered using the breadth first traversal. For example, after running Algorithm 3, in Fig. 1b of the online Appendix, node 10 is going to cover nodes 13 and 14 but it will not cover the nodes 1, 5, 6 an 11 as a part of this breadth first traversal. However, their covers will be assigned later by Algorithms 4 or 5 as appropriate.

5.5 Setting HopCount Values

We adopt a technique which helps in coverage by simple counting instead of using a plain breadth first search and thereby avoid covering already covered nodes redundantly. We introduce a notion of *hopCount*, which keeps track of how far a node is from an already selected influential node. When a node x is encountered while traversing upwards starting from another node y, we can check whether x has a non-zero *hopCount* value and thereby evaluate if the traversed part including y can be covered by an influential node that covers x. This method helps us in maximizing coverage of an already chosen influential node. Note that this trick acts as a substitute for reaching those nodes that could have been reached already if we did not restrict the BFS performed in Algorithm 3, in the upward direction or along the cross-edges. Nodes, which are assigned a *hopCount* value, can be primarily divided into two categories.

The first category is the predecessors of a just selected influential node v_i. An upward propagation of up to k hops is made from v_i to traverse at most k predecessors reachable using only tree-edges. The rth predecessor (located at r hops from v_i) is assigned a *hopCount* of r. If any of the predecessors p has a cross-edge (p, u) reaching an uncovered node u, then that cross-edge is also processed and u is also assigned a *hopCount* which is one more than the predecessor's *hopCount*. Note that there may be multiple cross-edges arising out of the same predecessor. From node u, once again an upward traversal is made to set *hopCount* values further for predecessors of u as long as the *hopCount* does not exceed k or the root is reached. In Fig. 1b of the online Appendix, while setting *hopCount*, node 10 finds a cross-edge from itself to node 11. Therefore, node 11 is assigned a *hopCount* value of 1. As the value of k is 2 in this example and node 11's *hopCount* does not exceed k's value, predecessors of node 11 are set *hopCount* values, without processing any further cross-edges from them. Like in Fig. 1b of the online Appendix, node 6 is assigned a *hopCount* value of two going upwards from node 11. As *hopCount* of node 6 becomes 2 (same as k), we stop further upward propagation from node 11. Upward propagation from node 10 continues and nodes 5 and 1 are assigned *hopCounts* of 1 and 2 respectively. When node 5 is reached, cross-edge e(5,6) is processed but the *hopCount* of node 6 is not updated because its *hopCount* via node 5 would not reduce its present *hopCount* value. Reduction of *hopCount* value of a node might allow it to cover more nodes from lower levels when it is encountered during an upward propagation for selection of an influential node and thereby save selection of an extra influential node.

The second category of nodes are the ones that can be reached via cross-edges from the descendants of v_i. Such a node can be reached via w, one of the descendants of v_i, where w can be reached from v_i only by tree-edges. After v_i is identified as an influential node, a k-hop breadth first traversal is run through the tree-edges linking v_i and its child nodes and their subsequent child nodes up to k levels. During this traversal, if any visited node is found to have a cross-edge, then the cross-edge is also processed and the node at the other end of the cross-edge (say u) is given a *hopCount* value. From u, its predecessors are also set *hopCount* values as long as the value does not exceed k or does not encounter root.

For another example illustrating the efficacy of setting *hopCount* values, in Fig. 1c of the online Appendix, had node 9 not been there in the graph, any of the nodes 2, 4 and 16 would have been selected as the starting point for upward propagation for selection of a new influential node. Any of those nodes would traverse upward by one hop to meet the root node having *hopCount* value 1. That means, those nodes can reach the node covering root node 15 in two hops. Hence, all three of those nodes would have been covered by node 3 and node 15 would not have been picked as a new influential node restricting the size of $|S|$.

Just as in the case of the tree algorithm, here also the black color is used for reduction of redundancy in traversal. When an influential node is selected and all the nodes in the levels below it are covered by traversal, it is colored black stating that any propagation through it would lead to nodes that are already covered.

In the above way, influential nodes are repeatedly selected to form the set of influential nodes S, until all the nodes of the graph are covered.

6 Empirical Validation

We now describe the context of applying the algorithm for finding influential researchers.

6.1 Experimental Setup

We have performed our experiments, including creating the network and running the k-hop dominating set finding algorithm on an Intel Xeon 2.4 GHz quad-core CPU desktop with 32 GB RAM and 500 GB hard disk. The operating system used for performing the experiments is Fedora LINUX version 3.3.4. The source code has been written in C++.

TABLE 2
Details of Data-Sets Used for Experiments

Domains	No of papers	No of unique authors	No of unique venues	Earliest record
Software engineering (SE)	23,779	14,347	16	1975
Operating System (OS)	4,348	3,177	15	1967
Databases (DB)	17,727	12,276	14	1960
Artificial intelligence (AI)	44,011	30,110	32	1960
Networking (NW)	18,741	12,507	13	1973
Total	108,606	72,417	90	-

6.2 Choice of Data-Sets

To validate our approach, we have used five data-set from different domains in the computing discipline. Table 2 outlines the contours of the data-sets. The choice of domains were based on the following considerations:

- *Software engineering*: SE has been transformed from being predominantly theoretical to a more empirical domain over the past few decades [50], [51]. As theoretical and empirical disciplines have different mores of collaboration [35], SE offers an interesting test-bed for exploring whether habits of successful researchers remain unchanged over the changing nature of a domain.
- *Operating System*: While research in operating systems has contributed several fundamental principles to computing, the OS community has remained largely close-knit and focused [52]. Thus, this community offers a distinct context for investigating our research question.
- *Databases*: Over the last fifty years, the DB community has addressed a wide range of research questions, from relational and other models of persisting and retrieving data, to the putative potential of big data [53]. In a sense, the DB community subsumes disciplines such as information retrieval, and data mining which have been popularized with the growing accessibility of data on the Web. How effective researchers operate in such a diverse community can illuminate our investigation.
- *Artificial intelligence*: Research in artificial intelligence began with immense promise in the middle of the 20th century, only to see much of its promise unfulfilled after several decades [54]. However, the recent resurgence of AI research has been significant, as noticed even by mainstream media [55]. Thus AI exemplifies a domain where research interest has periodically waxed and waned. This presents a very interesting context for examining our research question.
- *Networking*: Over the past few decades, the definition of a computer network has changed significantly with advent of the Internet, World Wide Web, and ubiquitous mobile computing devices. This has resulted in changing research paradigms in the NW community [56]. Including the NW domain in our study offers an opportunity to understand whether such change has also affected the habits of successful researchers.

With reference to Table 2, we note that the smallest and largest of our data-sets differ by more than an order of magnitude in the number of papers and unique authors (OS versus AI), while we cover more than fifty years of publication records (earliest records from 1960 in DB and AI). While there is always scope for examining additional domains, we believe these five domains offer sufficient size and variety for sound validation of our approach.

6.3 Data Collection, Processing and Network Generation

From the data-set provided by Aminer, we selected all the papers from a pre-defined set of venues (conferences and journals) for each of the SE, OS, DB, AI, NW domains. Implications of the choice of venues for each domain are discussed in Section 10. With the goal of forming a co-authorship network among the researchers in each domain, we started with a list of paper ID and author ID pairs, stating which paper was authored (or co-authored) by which author. From this list, we connected two researchers with an edge if they were found to be co-authors in at least one paper. This technique essentially forms a clique with all the co-authors of a paper. We did not include single author papers in this network as they would introduce self-loops. Single authorship was taken into account during our model development in Section 7.

For identifying influence in the co-authorship community, we selected only the giant component of the network for each domain and performed our analysis on the basis of the structure of the network. We assumed that due to the small number of researchers in the other components, they form a closed group of co-authors who may not be associated with the general collaboration ecosystem of a domain.

The network vertex parameters such as betweenness, clustering coefficient, and authority scores [57] used in our models (Section 7) were calculated using the giant component for each domain. The authority scores were calculated by the HITS algorithm [58]. In the interest of brevity, we show network visualizations for two domains in Fig. 2 of the online Appendix. The visualizations were generated using Gephi [59].

6.4 Implications of Network Type

As mentioned, we constructed co-authorship networks for our analysis. We recognize that other types of networks such as citation networks, or networks constructed on the basis of similarity of content can be used for related analysis. Each network type has distinct characteristics, and the choice of particular type carries its own implications. In scientific disciplines, researchers collaborate to augment their own ideas with new ones from collaborators [1]. Co-authorship is the principal vehicle for such collaboration. An instance of

TABLE 3
Cardinality of the kHDS Found by Our Algorithm for All k Values for the Author-Author Networks

k	SE	NW	OS	DB	AI
1	1,507	2,994	576	2,885	3,152
2	544	1,096	202	792	1,102
3	242	509	80	242	496
4	127	254	36	86	260
5	64	134	17	30	145
6	37	69	9	9	79
7	21	37	4	2	44
8	12	18	2	1	25
9	6	9	1	1	16
10	2	4	1	1	9

co-authorship is also an affirmation of conscious choice by the collaborators to share their ideas on a particular theme [34]. Citation networks can be biased by the widely recognized "reciprocity" effects [60], [61]. Networks based on similarity of paper content, will also be heavily dependent on the precision of automated techniques to detect such similarity [62]. Thus, we believe our choice of network type is most suitable for our context. We may point out that our general approach is agnostic of network type, and thus has applicability beyond co-authorship networks.

6.5 Running k-Hop Algorithm

We ran the k-hop algorithm on the giant components to find out the dominating sets for different values of k until the size of the dominating set became so small such that there could exist many such dominating sets of the same size. If the size of a k-hop dominating set (found by our algorithm) is denoted by $S(k)$, then for k values from 1 to 10, experimental results were as described in the Table 3. The k-hop algorithm took less than two minutes every time we ran it on our data-set for different values of k. We applied our algorithm on all the five data-sets.

7 Developing the Statistical Models

We now describe the development of statistical models to address our research question.

7.1 Computing Model Variables

After identifying the influential set of researchers as described above, we extracted a set of parameters for each researcher in each domain, for inclusion in the statistical models. These parameters were either readily available in the Aminer repository, or computed by database querying. To aid such querying, we had parsed relevant portions of the the Aminer repository into a MySQL database. The parameters extracted for each influential researcher in each domain were: the h-index [12] of the researcher, the number of publications of the researcher (*NoOfPublications*), the number of citations the researcher's publications have received (*NoOfCitations*), the number of single author papers the researcher has published (*NoOfSingleAuthorPapers*), the period of time (in years) over which the researcher has published papers in venues we considered (termed as *PublishingSpan*), the number of unique keywords in the researcher's papers (termed as *Diversity*), the number of papers written by the researcher as first author (termed as *NoOfLeadAuthorPapers*), the number of unique co-authors for the researcher (termed as *NoOfUniqueCo-Authors*). The other three parameters are network metrics: the betweenness centrality for a vertex-the number of shortest paths between all other vertices passing through it (termed as *Betweenness*), the clustering coefficient-signifies how many out of all the existing triplets have been converted into triangles, i.e., how many of the co-authorship possibilities have actually been converted to a co-author relationship (termed as *Collaboration*) [63] and authority score-depends on in-degree of the node and is calculated by a recursive definition using HITS algorithm [58] (termed as *Authority*).

7.2 Choice of Dependent Variable

The objective of our model is to understand the factors that relate to the impact of influential researchers. Our dependent variable, the *h-index* is a derived measure, that takes into account the number of publications as well as citations [12]. In our context the h-index is a more suitable measure than either of publication or citation count. Publication count does not take into account the reach or impact of the publications being counted, whereas citation count is often biased by the so called *Mathew effect* or preferential attachment [64], [63]. By its very definition, the h-index seeks to simultaneously reflect quantity and quality of an author's publications, and is relatively free from the biases of either publication or citation count. We recognize some of the common criticisms of h-index [65]. However, with reference to the discussion of Section 2, and the summary in Table 1, we believe h-index is the most suitable choice of the dependent variable in the context of this study.

7.3 Choice of Modeling Paradigm

For the five domains we are studying, we initially considered Poisson regression for modeling. Poisson distribution is defined by a single parameter, the mean, which is also equal to its variance. Overdispersion, which indicates a violation of the strong assumption of the equality of variance and mean, is a major threat to the validity of Poisson regression [66]. As this is present in our study, we initially modeled using negative binomial regression, which allows for over-dispersed count data by including an extra parameter in the model [67]. To check the stability of the outcome, we also modeled using multiple linear regression. The assumptions underlying multiple linear regression are linearity, normality, and homoscedasticity of the residuals, and absence of multicollinearity between the independent variables. The residual properties can be verified using histogram, Q-Q plot and scatter plot of the standardized residuals. We transformed some of our model variables by taking their square roots so that their distributions are closer to normal. The variance inflation factors (VIF) for the multiple linear regression model variables were within permissible limits; thus we inferred that multicollinearity did not pose a problem in our models.The overall significance of the models as well as the direction of effects did not change between negative binomial and multiple linear regression models. We chose to present the results from multiple linear regression modeling, as its outcome is more amenable to intuitive interpretation [68].

TABLE 4
Descriptive Statistics of the Model Variables for the SE Domain

Variable	Mean	SD	Median
h-index	4.413	5.072	3
NoOfPublications	22.285	36.759	8
NoOfCitations	194.329	558.465	34
NoOfSingleAuthorPapers	1.864	5.544	0
PublishingSpan	10.781	47.533	7
Diversity	9.658	1.502	10
*NoOfLeadAuthorPapers**	5.984	10.41	3
*NoOfUniqueCo-Authors**	26.512	38.335	12
*Betweenness**	1.796×10^{-4}	7.456×10^{-4}	0
Collaboration	0.695	0.391	1
*Authority**	1.078×10^{-4}	0.002	2.24×10^{-9}

Note: $*$ denotes the variable has been transformed in the regression model.

8 Results and Discussion

In this section, we present results from the regression models, discuss their implications for addressing the research question, and examine the suitability of the models for prediction.

8.1 Model Parameters

Table 4 of this paper and Tables 1, 3, 5, 7 of the online Appendix present the descriptive statistics for the SE, OS, DB, AI, and NW domains respectively. As specified in the table captions, some of the variables have been transformed for including in the regression models.

Table 5 of this paper and Tables 2, 4, 6, 8 of the online Appendix present the regression results for the SE, OS, DB, AI, and NW domains respectively. As our research question is concerned with identifying factors beyond publication and citation counts that relate to research impact, we have divided our model variables into the groups: *control variables* and *independent variables*. The model parameters help us discern the effects of the independent variables on the dependent variables, over and above the influence of the control variables. In the Table 5 of this paper and Tables 2, 4, 6, 8 of the online Appendix, the *base* model (column I) includes only the control variables, whereas the *refined* model (column II) additionally includes the independent variables. In these tables, along with the coefficient for each variable, we have indicated the standard error and the level of significance based on their respective p values, as specified in the tables' captions. The p value for each coefficient is calculated using the t-statistic and the Student's t-distribution. The lower parts of the tables give summary of the respective models: N is the number of data points used in modeling. R^2 is the coefficient of determination-the ratio of the regression sum of squares to the total sum of squares; it denotes the goodness-of-fit of a regression model. df denotes the degrees of freedom. F is the Fisher F-statistic-ratio of the variance in the data explained and the variance unexplained by the model. The p value for an entire model is calculated from the F-statistic and the F-distribution, it points to the overall statistical significance of the model. We can infer the corresponding result is statistically significant, if $p \leq$ *level of significance*, for the coefficients as well as the overall regression. The upper limit of the level of significance is taken as 0.05, on the basis of established practice [68].

As we observe from the Table 5 of this paper and Tables 2, 4, 6, 8 of the online Appendix, in each case, both the base and refined models are statistically significant, and including the independent variables over and above the control variable increases the the corresponding R^2 value. From the R^2 values, we observe that the goodness-of-fit of the refined models are

TABLE 5
Modeling the Impact of Influential Authors for the SE Domain

	I: Base Model			**II: Interaction Model**		
	Coefficient	Std error	Sig level	Coefficient	Std error	Sig level
Intercept	2.039	0.028	****	-0.601	0.129	****
Control variables						
NoOfPublications	0.073	9.751×10^{-4}	****	-0.003	0.001	**
NoOfCitations	0.004	6.418×10^{-5}	****	0.004	5.143×10^{-5}	****
Independent variables						
NoOfSingleAuthorPapers				0.354	0.021	****
PublishingSpan				6.353×10^{-4}	4.076×10^{-4}	-
Diversity				0.037	0.013	***
NoOfLeadAuthorPapers				0.473	0.019	****
NoOfUniqueCo-Authors				6.353×10^{-4}	4.076×10^{-4}	-
Betweenness				6.546	2.059	***
Collaboration				-0.322	0.056	****
Authority				-15.526	1.885	****
	Model parameters			**Model parameters**		
N	9,275			9,275		
R^2	0.793			0.871		
df	9,272			9,264		
F	1.773×10^4			6,229.578		
Sig level	****			****		

*Note: Significance levels "****", "***", "**", "*", "-", denote corresponding p-value $\leq 0.001, \leq 0.01, \leq 0.05, \leq 0.1$, and ≥ 0.1 respectively.*

around 87, 90, 87, 86, and 87 percent for the SE, OS, DB, AI, and NW domains respectively. Looking at the coefficients of the model variables, their signs and levels of significance for the refined model for each domain, we are able to understand how the independent variables relate to the dependent variable (*h*-index). Table 7 summarizes these influences; the following discussion refers to this table.

8.2 Influence of the Models Variables

Let us consider the control variables first. The number of publications has a statistically significant effect on impact across all domains, other than OS. More publications relate to higher impact for DB, AI, and NW, and to lower impact for SE and OS. More citations relate to higher impact across all domains and all the relations are statistically significant. While it is expected that increasing number of publications and citations makes it more likely for a researcher to have a higher impact; we see evidence that the nature of the SE and OS disciplines makes it unlikely that more publications would indicate higher impact.

Turning now to the independent variables, we observe that more single author papers relate to higher impact across all domains, with statistical significance (other than OS). Single author papers reflect mastery of the paper's topic, and higher level of mastery can be expected to associate with higher impact. The effect of *PublishingSpan* is statistically significant for all domains other than SE. Higher publishing span, indicating a longer duration of active research relates to higher impact for SE, OS, DB, and AI; but it relates to lower impact for NW. Given the changes in research directions in NW with the advent of the Web and mobile computing, older results may have been subsumed by newer ones. We see higher diversity relating to higher impact across all domains, but the effect is statistically significant only for SE and AI. Thus, casting a wider net of research interests seems to facilitate higher impact for researchers across domains. In this day and age of increasing specialization, this evidence offers a fresh perspective into how researcher attention can be effectively focused. More papers as lead authors point to a strong leadership position for a researcher, and it is natural that such a position would translate to higher impact. We see congruent empirical evidence; higher *NoOfLeadAuthorPapers* relates to higher impact across all domains, and all effects are statistically significant. We observe that higher number of unique co-authors relate to higher impact across all domains other than NW (the effect is statistically significant for OS, DB, and AI). Co-authors bring in new ideas and can serve as important connections in the research ecosystem. Also, co-authors are more aware of one another's work, which may facilitate more citations [69]. Thus it is likely that higher levels of connections will relate to higher impact. Further investigation is required to explain why the trend is opposite for NW. Higher betweenness is seen to relate to higher impact in SE, OS, DB, AI; and the effect is statistically significant for SE, OS, DB. While the effect is inverse (though not statistically significant) for NW. Lying on many of the shortest paths between other researchers in a collaboration network signifies a position of eminence for a researcher; and it is expected such a position would come with higher impact. Interestingly, we observe that higher collaboration as well as higher authority are associated with *lower* impact across all domains and both the effects are statistically significant across all domains, barring only authority, for AI. These are notable and counter-intuitive results which warrant a closer look.

Collaboration is one of the buzzwords of our times. With transformational changes in communication technologies over the past two decades, contact between peers is now easy and pervasive. The effects of such contact are widespread in different areas, and the world of research is no exception. The number of single author papers in areas of computing has steadily declined over the years [70]; with changing nature of the discipline, it is becoming a norm to have several authors in a paper [71]. But are researchers enriched by collaboration, specially in the computing disciplines we are examining?

Brooks has passionately argued that great artifacts very often come from the confluence of very few minds [72]. Dijkstra pointed out how over-emphasizing the importance of communication, brings with it "standard pressures of conformity"; he even credited his isolation with fostering the development of original ideas.[3] In our study, evidence of higher collaboration relating to lower impact is likely to be an indication of the challenges Brooks and Dijkstra have commented upon. If further studies in other disciplines within and without computer science reveal a similar relationship, it can lead to a rethinking of the dynamics of research at the individual and organizational levels.

Berlin's essay *The Hedgehog and the Fox*, uses a zoological metaphor to explore an interesting dichotomy in the ways of human thinking [73]. The title is purportedly taken from the ancient Greek poet Archilochus' saying "a fox knows many things, but a hedgehog one important thing". Accordingly, Berlin classifies thinkers into two categories: "hedgehogs" pursue a single central theme, whereas "foxes" explore a variety of ideas. Although Berlin concerned himself with individuals whose impact on society can not merely be measured by bibliometric indices,[4] his classification is relevant to reflective activities such as research. Unknowingly, researchers often have to grapple with the hedgehog versus fox question at various stages in their careers: whether to focus exclusively on one or a small set of closely related ideas, or pursue a "portfolio" of varied themes. In our study, we find evidence that higher diversity relates to higher impact, whereas higher authority relates to lower impact. Authority is indicative of deep and prolonged attention, whereas diversity implies wider and perhaps more short-lived interests. Our results point to foxes rather than hedgehogs having higher impact!

8.3 Prediction Accuracy

To evaluate whether and how our refined regression models are suitable for predicting the *h*-index, we present several metrics in Table 6. As we observe, the correlation (Correl) between the predicted and actual values of the *h*-index is more than 0.9 for all domains, the mean absolute errors (MAE) are very low (the highest being of the 10^{-16} order of magnitude), the root mean square errors (RMSE) are also low, and when they are normalized by the range of the *h*-index for

3. http://cacm.acm.org/magazines/2010/8/96632-an-interview-with-edsger-w-dijkstra/fulltext

4. For a list of Berlin's hedgehogs and foxes, see https://en.wikipedia.org/wiki/The_Hedgehog_and_the_Fox

TABLE 6
Prediction Accuracy in Terms of Correlation Between Actual and Predicted Values of the Independent Variable(Correl), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), RMSE Normalized by the Range of the Independent Variable (NRMSE), Median of 10-Fold Cross Validation Mean Squared Error (XVALID)

Domains	Correl	MAE	RMSE	NRMSE	XVALID
Software engineering (SE)	0.933	-6.851×10^{-16}	1.825	0.03	3.19
Operating system (OS)	0.952	5.792×10^{-18}	2.174	0.039	4.805
Databases (DB)	0.941	-1.057×10^{-16}	1.752	0.029	3.05
Artificial Intelligence (AI)	0.925	2.935×10^{-18}	1.769	0.033	3.235
Networking (NW)	0.933	-1.018×10^{-17}	1.491	0.025	2.225

each domain (NRMSE) are also around 0.03 for all domains. To further understand prediction accuracy, we performed 10-fold cross validation. The median of the mean squared errors across all 10 folds was found to be less than 5 for all domains. These results lead us to infer that the refined regression models can be used for predicting the h-index of individual researchers with reasonably high accuracy.

In subsequent sections, we highlight how our results can be put to use, followed by a discussion of the threats to their validity.

9 Utility of the Results

Our algorithm for identifying influential authors, methodology for conducting the study, and insights from our results can be useful in the following ways:

1) For academic and industrial researchers, research is a high stakes game. Not only are promotions and tenure decisions dependent on its outcome, every researcher is also interested in influencing his/her community. Our empirical evidence can serve as a rubric for researchers in planning and executing their research agenda. The insights on factors beyond publication and citation counts that relate to research impact, will help researchers make informed choices on the questions they investigate, the papers they write, and the peers they collaborate with. These choices are customarily influenced by perceptions, hearsay, and advice; our results will help researchers decide for themselves on the face of empirical evidence.

TABLE 7
Directionality of Influence of Model Variables on Dependent Variable

Model variable	SE	OS	DB	AI	NW
NoOfPublications	↙	↙	↗	↗	↗
NoOfCitations	↗	↗	↗	↗	↗
NoOfSingleAuthorPapers	↗	↗	↗	↗	↗
PublishingSpan	↗	↗	↗	↗	↙
Diversity	↗	↗	↗	↗	↗
NoOfLeadAuthorPapers	↗	↗	↗	↗	↗
NoOfUniqueCo-Authors	↗	↗	↗	↗	↙
Betweenness	↗	↗	↗	↗	↙
Collaboration	↙	↙	↙	↙	↙
Authority	↙	↙	↙	↙	↙

*Note: ↗ (in **bold** font) denotes dependent variable* increases *with increase in model variable; ↙ (in* bold *font) denotes dependent variable* decreases *with increase in model variable; ↗ and ↙ denote corresponding relations which are not statistically significant.*

2) The k-hop dominating set-based influential node finding algorithm can be used in a viral marketing scenario very effectively. Often a viral marketing campaign is constrained by limited funding or by urgency of the campaign. The fastest way to reach everyone is to provide incentive to everyone, i.e., every node in the network, so that they could accept the information. No traversal is needed to spread the information. However, if the funding is limited, then it is not possible any more to provide incentive to everyone. In that case, it is essential for the marketing campaign to compromise on the speed of the spread and select a smaller starting set of nodes, who will initiate the spread of information. Such selections can also be made by the k-hop dominating set finding algorithm, because the algorithm helps generate a small sized k-hop dominating set keeping the marketing expense within the limit of the budget. Increase in k will lead to a smaller set of starting nodes, thereby requiring a smaller budget. If the calculated budget comes out to be more than the allocated budget for a value of k, then k can be increased by a step of 1 to see whether the revised budget comes down to the allocated budget.
3) The refined statistical models across different domains can be used to predict h-indices for researchers. Such prediction helps calibrate research productivity at the individual and organizational levels.

10 Threats to Validity and Future Work

This is an observational study, rather than a controlled experiment. Thus correlation does not necessarily imply causation in our results. To establish a cause-and-effect relationship between the independent and dependent variables, a controlled experiment is required. Given the spontaneous nature of research, such an experiment is difficult to set up and execute. Thus we believe our results offer interesting insights within the inherent limitations of a observational study. We discuss below other limitations of our study with respect to *construct validity*, *internal validity*, *external validity* and *reliability*.

Construct validity is concerned with correct measurement of the variables. Representation of our model variables such as betweenness, collaboration and authority are based on established network metrics [63]. We have chosen to consider h-index as a proxy for research impact. While we recognize the lack of consensus on how best to measure impact in research [74], our selection of h-index aligns with its wide

usage [7]. If a different measure of impact is chosen, the results can be different.

A study manifests *internal validity* if it is free from systematic errors and biases. A concern related to the use of citation count is the accuracy of the raw citation data, which can be mitigated by concerted efforts at cleaning and consistency checking [75], [76]. Our use of the Aminer data-sets in this study helps address these concerns. Since we used the Aminer data-sets, issues that can affect internal validity such as mortality (subjects withdrawing from a study during data collection) and maturation (subjects changing their characteristics during the study outside the parameters of the research) do not arise in our case. However, our selection of publication venues for the SE, OS, DB, AI, NE domains can be a source of bias. Although we believe our corpora for this study covers a major portion of each domain's research publications, we cannot claim to have captured *all* published papers in the time period we have studied. Whether or not a particular venue included in our study is a purely SE, OS, DB,AI, or NW venue is a matter of judgment. With considerable overlap among computing disciplines, it is likely we have considered some papers which relate to a particular domain only in a broad sense, and we may have inadvertently excluded some papers from a particular domain. Given the sizes of our corpora, we do not believe these issues will significantly alter our results. A usual problem in studies of scientific publication arises out of the ambiguity of author names. If the same author has been differently identified as "John Doe", "J Doe" etc. in different publications, it is very hard to reconcile their identities without manual intervention. Conversely, if there are multiple individuals called "John Doe", it is difficult to distinguish them. Such inconsistencies are minimized in DBLP through significant human intervention [77]. We are not sure to what extent such intervention has been carried out in the Aminer database. We detected such issues in a small percent of author names in our corpora; the conflicting author names were manually removed.

External validity relates to the generalisability of results. Even as our sample size and sampling method are unlikely to be a threat to external validity, and we have studied five different domains, we do not claim our results to be generalizable as yet across disciplines. Every discipline has its own mores on what is considered as influential research, and an influential researcher in one discipline may not hold a similar position in another. Further studies are required to confirm the validity of our results in other disciplines.

Reliability of a study is associated with the reproducibility of the results. As there is minimal human intervention in running the steps of our study, given access to the data-sets, our results can be reproduced easily.

In our plans for *future work*, we seek an understanding of the influential nodes by network topological analysis and predicting future co-authorship possibilities by link prediction. We also plan to extend this study across other disciplines.

11 Summary and Conclusions

In this paper we extracted corpora of research publications in the domains of software engineering, operating system, databases, artificial intelligence, and networking from the Aminer repository and constructed co-authorship networks from the data. We presented and applied a novel algorithm to identify a set of influential researchers from the networks. A statistical analysis of the factors influencing research impact of these researchers revealed a number of insights. Within the scope of our study and with some exceptions, we found evidence that irrespective of the domain, influential researchers with high impact write more single author papers, remain active in publishing for a longer period of time, publish on a wide variety of topics, are lead authors in many of their papers, have many unique co-authors; but they collaborate less, and are in lower positions of authority in their peer groups. These results can inform individual and collective decision making in research enterprises, as well as stimulate a discussion around how research impact is measured, and how individual researchers attain high impact.

References

[1] T. S. Kuhn, *The Structure of Scientific Revolutions*, 3rd ed. Chicago, IL, USA: Univ Chicago Press, Dec. 1996.

[2] K. Börner, L. Dall'Asta, W. Ke, and A. Vespignani, "Studying the emerging global brain: Analyzing and visualizing the impact of co-authorship teams," *Complexity*, vol. 10, pp. 57–67, 2005.

[3] D. L. Parnas, "Stop the numbers game," *Commun. ACM*, vol. 50, no. 11, pp. 19–21, Nov. 2007.

[4] S. N. Dorogovtsev and J. F. F. Mendes, "Evolution of networks," *Advance Phys.*, vol. 51, pp. 1079–1187, 2002.

[5] J. McKeen Cattell and D. R. Brimhall, *American Men of Science; A Biographical Directory*. Garrison, NY, USA: Sci. Press, 1921.

[6] I. H. Sher and E. Garfield, "New tools for improving and evaluating the effectiveness of research," in *Proc. Res. Program Effectiveness*, 1966, pp. 27–29.

[7] R. van Noorden, "Metrics: A profusion of measures," *Nature News*, vol. 465, no. 7300, pp. 864–866, Jun. 2010.

[8] J. Bollen, H. van de Sompel, A. Hagberg, and R. Chute, "A principal component analysis of 39 scientific impact measures," *PLoS One*, vol. 4, no. 6, Jun. 2009, Art. no. e6022.

[9] Colin Macilwain, "World view: Wild goose chase," *Nature News*, vol. 463, no. 7279, pp. 291–291, Jan. 2010.

[10] A. Abbott, D. Cyranoski, N. Jones, B. Maher, Q. Schiermeier, and R. Van Noorden, "Metrics: Do metrics matter?" *Nature News*, vol. 465, no. 7300, pp. 860–862, Jun. 2010.

[11] D. De Solla Price, *Little Science, Big Science... and Beyond*. New York, NY, USA: Columbia Univ. Press, Aug. 1986.

[12] J. E. Hirsch, "An index to quantify an individual's scientific research output," *Proc. Nat. Academy Sci. United States America*, vol. 102, no. 46, pp. 16569–16572, Nov. 2005.

[13] P. Ball, "Achievement index climbs the ranks," *Nature*, vol. 448, no. 7155, pp. 737–737, Aug. 2007.

[14] D. Butler, "Web usage data outline map of knowledge," *Nature News*, vol. 458, Mar. 2009, Art. no. 135.

[15] J. Lane, "Let's make science metrics more scientific," *Nature*, vol. 464, no. 7288, pp. 488–489, Mar. 2010.

[16] T. Braun, et al., "How to improve the use of metrics," *Nature*, vol. 465, no. 7300, pp. 870–872, Jun. 2010.

[17] H. Tong and C. Faloutsos, "Center-piece subgraphs: Problem definition and fast solutions," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 404–413.

[18] H. Tong, C. Faloutsos, B. Gallagher, and T. Eliassi-Rad, "Fast best-effort pattern matching in large attributed graphs," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2007, pp. 737–746.

[19] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social Netw.*, vol. 1, pp. 215–239, 1978.

[20] P. Bonacich, "Power and centrality: A family of measures," *Amer. J. Sociology*, vol. 92, no. 5, pp. 1170–1182, 1987.

[21] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," *Comput. Netw. ISDN Syst.*, vol. 30, no. 1–7, pp. 107–117, Apr. 1998.

[22] A. Bavelas, "Communication patterns in task-oriented groups," *J. Acoustical Soc. America*, vol. 22, no. 6, pp. 725–730, Nov. 1950.

[23] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, vol. 40, no. 1, pp. 35–41, Mar. 1977.

[24] F. Wang, et al., "On positive influence dominating sets in social networks," *Theoretical Comput. Sci.*, vol. 412, no. 3, pp. 265–269, Jan. 2011.

[25] T. N. Dinh, Y. Shen, D. T. Nguyen, and M. T. Thai, "On the approximability of positive influence dominating set in social networks," *J. Combinatorial Optimization*, vol. 27, no. 3, pp. 487–503, Apr. 2014.

[26] J. C. Nacher and T. Akutsu, "Dominating scale-free networks with variable scaling exponent: Heterogeneous networks are not difficult to control," *New J. Phys.*, vol. 14, no. 7, 2012, Art. no. 073005.

[27] C. Borgs, M. Brautbar, J. Chayes, S. Khanna, and B. Lucier, "The power of local information in social networks," in *Proc. 8th Int. Workshop Internet Net. Econ.*, 2012, pp. 406–419.

[28] F. Molnar Jr., S. Sreenivasan, B. K. Szymanski, and G. Korniss, "Minimum dominating sets in scale-free network ensembles," *Sci. Rep.*, vol. 3, 2013, Art. no. 1736.

[29] P. Basuchowdhuri and S. Majumder, "Finding influential nodes in social networks using minimum k-hop dominating set," in *Applied Algorithms*. Berlin, Germany: Springer, pp. 137–151.

[30] S. Kundu and S. Majumder, "A linear time algorithm for optimal k-hop dominating set of a tree," *Inf. Process. Lett.*, vol. 116, no. 2, pp. 197–202, 2015.

[31] X. Zhu, J. Yu, W. Lee, D. Kim, S. Shan, and D. Z. Du, "New dominating sets in social networks," *J. Global Optimization*, vol. 48, no. 4, pp. 633–642, 2010.

[32] S. Guha and S. Khuller, "Approximation algorithms for connected dominating sets," *Algorithmica*, vol. 20, pp. 374–387, 1998.

[33] A. L. Barabasi, H. Jeong, Z. Neda, E. Ravasz, A. Schubert, and T. Vicsek, "Evolution of the social network of scientific collaborations," *Physica A*, vol. 311, no. 3/4, pp. 590–614, 2002.

[34] M. E. J. Newman, "The structure of scientific collaboration networks," *Proc. Nat. Academy Sci. United States America*, vol. 98, pp. 404–409, 2000.

[35] M. E. J. Newman, "Scientific collaboration networks. I. Network construction and fundamental results," *Phys. Rev. E*, vol. 64, no. 1, 2001, pp. 016131.

[36] M. E. J. Newman, "Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality," *Phys. Rev. E*, vol. 64, no. 1, 2001, Art. no. 016132.

[37] M. E. J. Newman and J. Park, "Why social networks are different from other types of networks," *Phys. Rev. E*, vol. 68 May 2003, Art. no. 036122.

[38] L. M. A. Bettencourt, D. I Kaiser, J. Kaur, C. Castillo-Chvez, and D. E. Wojick, "Population modeling of the emergence and development of scientific fields," *Scientometrics*, vol. 75, no. 3, pp. 495–518, 2008.

[39] M. Herrera, D. C. Roberts, and N. Gulbahce, "Mapping the evolution of scientific fields," *PLoS One*, vol. 5, no. 5, May 2010, Art. no. e10355.

[40] S. Gerrish and D. M. Blei, "A language-based approach to measuring scholarly impact, in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 375–382.

[41] J. Huang, Z. Zhuang, J. Li, and C. L. Giles, "Collaboration over time: Characterizing and modeling network evolution," in *Proc. Int. Conf. Web Search Web Data Mining*, 2008, pp. 107–116.

[42] A. E. Hassan and R. C. Holt, "The small world of software reverse engineering," in *Proc. 11th Working Conf. Reverse Eng.*, 2004, pp. 278–283.

[43] R. L. Glass, I. Vessey, and V. Ramesh, "Research in software engineering: An analysis of the literature," *Inf. Softw. Technol.*, vol. 44, no. 8, pp. 491–506, 2002.

[44] R. L. Glass, V. Ramesh, and I. Vessey, "An analysis of research in computing disciplines," *Commun. ACM*, vol. 47, pp. 89–94, Jun. 2004.

[45] E. Ahmed, A. Gani, M. Sookhak, S. H. Ab Hamid, and F. Xia, "Application optimization in mobile cloud computing," *J. Netw. Comput. Appl.*, vol. 52, no. C, pp. 52–68, Jun. 2015.

[46] E. Ahmed, A. Gani, M. K. Khan, R. Buyya, and S. U. Khan, "Seamless application execution in mobile cloud computing: Motivation, taxonomy, and open challenges," *J. Netw. Comput. Appl.*, vol. 52, pp. 154–172, Jun. 2015.

[47] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. U. Khan, "The rise of big data on cloud computing: Review and open research issues," *Inform. Syst.*, vol. 47, pp. 98–115, Jan. 2015, http://dx.doi.org/10.1016/j.is.2014.07.006

[48] I. A. T. Hashem, et al., "The role of big data in smart city," *Int. J. Inf. Manage.*, vol. 36, no. 5, pp. 748–758, 2016.

[49] I. Yaqoob, et al., "Information fusion in social big data: Foundations, state-of-the-art, applications, challenges, and future research directions," *Int. J. Inf. Manage.*, 2016.

[50] M. Shaw, "Prospects for an engineering discipline of software," *IEEE Softw.*, vol. 7, no. 6, pp. 15–24, Nov. 1990.

[51] M. Shaw, "Continuing prospects for an engineering discipline of software," *IEEE Softw.*, vol. 26, no. 6. pp. 64–67, Nov./Dec. 2009.

[52] P. J. Denning, "Fifty years of operating systems," *Commun. ACM*, vol. 59, no. 3, pp. 30–32, Feb. 2016.

[53] D. Abadi, et al., "The Beckman report on database research," *ACM SIGMOD Rec.*, vol. 43, no. 3, pp. 61–70, Dec. 2014.

[54] L. E. Travis, "In defense of artificial intelligence research," *Commun. ACM*, vol. 5, no. 1, pp. 6–7, Jan. 1962.

[55] B. Hodjat, "The AI resurgence: Why now?" 2015. [Online]. Available: http://www.wired.com/insights/2015/03/ai-resurgence-now/, Accessed on: May 21, 2016.

[56] C. Partridge, "Forty data communications research questions," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 5, pp. 24–35, Oct. 2011.

[57] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 409–10, 1998.

[58] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *J. ACM*, vol. 46, no. 5, pp. 604–632, 1999.

[59] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: An open source software for exploring and manipulating networks," in *Proc. Int. AAAI Conf. Weblogs Social Media*, 2009, pp. 361–362.

[60] E. Fehr and S. Gächter, "Fairness and retaliation: The economics of reciprocity," *J. Econ. Perspectives*, vol. 14, no. 3, pp. 159–181, 2000.

[61] L. Stanca, "Measuring indirect reciprocity: Whose back do we scratch?" *J. Econ. Psychology*, vol. 30, no. 2, pp. 190–202, 2009.

[62] D. Blei, A. Ng, and M. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[63] M. E. J. Newman, "The structure and function of complex networks," *SIAM Rev.*, vol. 45, pp. 167–256, Mar. 2003.

[64] D. J. de Solla Price, "Networks of scientific papers," *Science*, vol. 149, no. 3683, pp. 510–515, 1965.

[65] M. C. Wendl, "H-index: However ranked, citations need context," *Nature*, vol. 449, no. 7161, pp. 403–403, Sep. 2007.

[66] D. N. Barron, "The analysis of count data: Overdispersion and autocorrelation," *Sociological Methodology*, vol. 22, pp. 179–220, 1992.

[67] A. C. Cameron and P. K. Trivedi, *Regression Analysis of Count Data*. Cambridge, U.K.: Cambridge Univ. Press, 1998.

[68] B. G. Tabachnick and L. S. Fidell, *Using Multivariate Statistics*. Boston, MA, USA: Pearson Education, 2007.

[69] S. Datta, N. Kumar, and S. Sarkar, "The social network of software engineering research," in *Proc. 5th India Softw. Eng. Conf.*, 2012, pp. 61–70.

[70] S. Datta, S. Sarkar, A. S. M. Sajeev, and N. Kumar, "How many researchers does it take to make impact?: Mining software engineering publication data for collaboration insights," in *Proc. 6th ACM India Comput. Convention*, 2013, pp. 6:1–6:8.

[71] M. E. J Newman, "Clustering and preferential attachment in growing networks," *Phys. Rev. E*, vol. 64, 2001, Art. no. 025102.

[72] F. P Brooks, *The Design of Design: Essays from a Computer Scientist*. Upper Saddle River, NJ, USA: Addison-Wesley, 2010.

[73] I. Berlin, *The Hedgehog and the Fox: An Essay on Tolstoy's View of History*. Chicago, IL, USA: Ivan R. Dee, Publisher, 1993.

[74] J. Grudin, "Technology, conferences, and community," *Commun. ACM*, vol. 54, no. 2, pp. 41–43, Feb. 2011.

[75] D. Adam, "Citation analysis: The counting house," *Nature*, vol. 415, no. 6873, pp. 726–729, Feb. 2002.

[76] D. Butler, "Academics strike back at spurious rankings," *Nature*, vol. 447, no. 7144, pp. 514–515, May 2007.

[77] M. Ley and P. Reuther, "Maintaining an Online Bibliographical Database: The problem of data quality," *EGC*, vol. RNTI-E-6, pp. 5–10, 2006.

Subhajit Datta received the PhD degree in computer science from Florida State University. He is currently a lecturer with Singapore University of Technology and Design. He has more than 17 years of experience in software design, development, research, and teaching with various organizations in the US, India, and Singapore. He is the author of the books *Software Engineering: Concepts and Applications* (Oxford University Press, 2010) and *Metrics-Driven Enterprise Software Development* (J. Ross Publishing, 2007), which are widely used by students and practitioners. His research interests include software architecture, empirical software engineering, social computing, and big data. More details about his background and interest are available at www.dattas.net.

Partha Basuchowdhuri received the bachelor's degree in electronics and telecommunication from Bengal Engineering College (presently known as IIEST, Shibpur), the master's degrees, one in computer science and the other in electronics and electrical engineering, from Louisiana State University (LSU), Baton Rouge, and the PhD degree from Jadavpur University, in March 2016. He is an assistant professor in the Department of Computer Science and Engineering, Heritage Institute of Technology, Kolkata. His primary research interests include social network analysis and data mining. His recent works have been accepted for publication in foreign conference venues like PAKDD, ASONAM, ACM IKDD CoDS, ICMLA, and IEEE GrC. His master's degree thesis work at LSU was funded by US National Science Foundation. After completing the master's degree he moved to Digital Enterprise Research Institute (DERI), Ireland to continue his research in social network analysis. He briefly worked as a post-doctoral fellow with Queen's University Belfast from April, 2016 to August, 2016. He is also the founding faculty sponsor of the ACM student chapter with Heritage Institute of Technology.

Surajit Acharya received the MTech degree from the Department of Computer Science and Engineering, Birla Institute of Technology, Bits, Mesra, India, and the MCA degree from Sikkim Manipal University. He is working as a technical assistant in the Department of Computer Science and Engineering, Heritage Institute of Technology, Kolkata, West Bengal. His research interests include the area of data mining, data analysis, empirical software engineering, etc.

Subhashis Majumder received the MTech degree in computer science from the Indian Statistical Institute and the PhD degree in computer science and engineering from Jadavpur University, Kolkata. He also worked as a research assistant in the Computer Engineering Department, Rutgers University, for a year. After finishing his undergraduate work in the Electronics and Telecommunication Engineering Department, Jadavpur University, Kolkata, in 1993, he started his career with Texas Instruments India Pvt. Ltd. and has got more than seven years of industry experience. He has led product development teams working on innovative cutting edge technologies and has the experience of working as the technology head of a medium-sized IT Company. In 2003, he had shifted to full-time academics and is presently serving as a full professor and head of the CSE Department of Heritage Institute of Technology (HIT), Kolkata. He is also the dean of UG Programs with HIT. He has more than 50 publications in international conferences as well as archival journals. His research interests include graph algorithms, physical design algorithms, algorithmic microfluidics, algorithms for social networking, and empirical software engineering. He has been received an award for excellence in the category of outstanding research accomplishment at HIT in 2016. He also received an NTS scholar (1987) by NCERT, India. In 2006, he had founded a small knowledge-based IT firm named Ditsa Technologies that focused on niche projects requiring algorithm-intensive programming techniques. His key contributions there were some fast and efficient scheduling algorithms based on which the key projects evolved.