Singapore Management University

Institutional Knowledge at Singapore Management University

Research Collection School Of Computing and Information Systems

School of Computing and Information Systems

12-2021

Degree doesn't matter: Identifying the drivers of interaction in software development ecosystem

Amrita BHATTACHARJEE
Heritage Institute of Technology

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

Subhashis MAJUMDER
Heritage Institute of Technology

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research

Part of the Software Engineering Commons

Citation

BHATTACHARJEE, Amrita; DATTA, Subhajit; and MAJUMDER, Subhashis. Degree doesn't matter: Identifying the drivers of interaction in software development ecosystem. (2021). *2021 Asia-Pacific Software Engineering Conference 28th APSEC: Taipei, Virtual, December 6-9: Proceedings*. 410-420.
Available at: https://ink.library.smu.edu.sg/sis_research/6581

This Conference Proceeding Article is brought to you for free and open access by the School of Computing and Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Computing and Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email cherylds@smu.edu.sg.

Degree doesn't Matter: Identifying the Drivers of Interaction in Software Development Ecosystems

Ishita Bardhan
Dept of Computer Sc. & Engg.
Heritage Institute of Technology
Kolkata, India
ishita.bardhan.cse21@heritageit.edu.in

Subhajit Datta
School of Computing & Information Systems
Singapore Management University
Singapore
subhajit.datta@acm.org

Subhashis Majumder
Dept of Computer Sc. & Engg.
Heritage Institute of Technology
Kolkata, India
subhashis.majumder@heritageit.edu

***Abstract*—Large scale software development ecosystems represent one of the most complex human enterprises. In such settings, developers are embedded in a web of shared concerns, responsibilities, and objectives at individual and collective levels. A deep understanding of the factors that influence developers to connect with one another is crucial in appreciating the challenges of such ecosystems as well as formulating strategies to overcome those challenges. We use real world data from multiple software development ecosystems to construct developer interaction networks and examine the mechanisms of such network formation using statistical models to identify developer attributes that have maximal influence on whether and how developers connect with one another. Our results challenge the conventional wisdom on the importance of particular developer attributes in their interaction practices, and offer useful insights for individual developers, project managers, and organizational decision-makers.**

***Index Terms*—software development ecosystems, ERGM, degree, closeness, pagerank**

I. Introduction

Successful functioning of large scale software development ecosystems are underpinned by effective interaction between developers. Interactions between developers can offer vital conduits for the flow of information and experience in a project team. Naturally, facilitating the setting up and sustenance of such helpful connections is a key concern for project managers. Thus understanding the drivers of "who connects to whom" has strong implications for individual developer contributions as well team assembly and governance.

As with many other collective human enterprise [1], a software development ecosystem can be modelled as a network whose nodes represent developers, and two developers are connected by a directed or undirected link on the basis of some conjoint interest or activity [2]. The growth and evolution of such networks have been studied in depth across different domains and models have been established to explain their key dynamics [3]. We may apply such models in the context of large scale software development. For example, a software development ecosystem can be thought of as a growing random network where a new node is born over time and attaches itself to an existing node; when the incoming new node randomly selects a node for connection, this can be abstracted as the growing variation of the Erdos Renyi model [4]. As another example, the preferential attachment model [5] when applied to the software development context will mandate nodes who already have many connections (higher degree), to attract more incoming nodes.

While these and other existing models have wide relevance in varied scenarios, we believe that they ignore a vital characteristic of large scale software development. Predominantly, existing models assume that single attribute of a node such as its degree is the only property that influences how it will be connected to other nodes. This can be a valid assumption when nodes represent something like a page on the World Wide Web, with degree denoting the number of hyperlinks to other Web pages. However in a complex enterprise such as software development with its involved needs of knowledge, experience, and expertise, a developer's degree can only capture one of his/her multiple facets in the network representing the development ecosystem. Accordingly, we posit that a deep understanding of what drives developer connections can only be understood if the influences of multiple attributes are considered in the link formation process. With this background, we examine the following **research question** in this paper: *In large scale software development ecosystems, which developer attribute(s) maximally influence developer interaction?*

Addressing this question enables our study to make the following research contributions:

- We present a methodology for identifying factors that influence developer interaction in large scale software development ecosystems.
- We underscore the need to look at the multi-faceted aspects of developer interaction vis-a-vis considering such interaction to be influenced by any single factor.
- We identify the key drivers of developer interaction and establish that our results hold across multiple interaction networks and software development ecosystems.

In the next section, we present an outline of related work, followed by a description of our study setting. Subsequently, we explain our methodology, discuss our results and identify threats to their validity. The paper ends with a summary, and conclusions from the study.

II. Related Work

A. Network models

Real world networks have heavy tailed degree distributions, small diameters, and high levels of clustering. Additionally, many such networks in nature and society are scale-free along with high clustering. Ravasz and Barabasi [6] show that these two features are the consequence of a hierarchical organization, implying that small groups of nodes organize in a hierarchical manner into increasingly large groups, while maintaining a scale-free topology. Dorogovtsev et al. [7] propose a deterministic network model which obeys the scaling law between the node degree and clustering coefficient, in addition to the power-law degree distribution. Comellas et al. [8] discuss a category of graphs – recursive clique trees – which have small-world and scale-free properties and allow a fine tuning of the clustering and the power-law exponent of their discrete degree distribution. Chen et al. [9] have introduced a family of planar, modular and self-similar graphs which have small-world and scale-free properties but all nodes have zero clustering coefficient. This model with a null clustering coefficient can be used to represent networks with small clustering coefficient and can be used to study other properties without the influence of clustering. Song et al. [10] show that complex networks have self similar structures. Golnari and Zhang [11] propose a multivariate analysis perspective to study complex structures in networks.

Jackson introduced a special class of models – hybrid models – where newly born nodes are linked to some nodes uniformly at random and to some other nodes by navigating through the network [4] . Even in his meeting based network formation model, each new node "meets" some number of nodes uniformly at random forming direct links to them, and then chooses some of the out links from the first group of nodes and follows them to meet new nodes and form additional links [12].

B. Link formation

Nguyen et al. [13] propose a general framework to define link formation behaviours using well studied local link structures (i.e. triads and dyads) in a dynamic social network where links are formed at different timestamps. They find that these behaviours become more stable as the users establish more links. Leskovec et al. [14] studied the individual node arrival and link creation processes that collectively lead to macroscopic properties of networks. Their findings suggest that link locality play a critical role in evolution of networks. Leskovec et al. [15] also observed some surprising phenomena: real world networks become more dense over time with the number of links growing super linearly with the number of nodes; and the average distance between nodes in such networks often shrinks over time. Link formation (LF) has been studied from different perspectives in the analysis of social networks. Leung et al. [16] propose the approach of mining interesting LF rules containing link structures known as LF-patterns. LF-rules capture the formation of new link from a focal node to another node as a post-condition of existing connections between the two nodes. Bahulkar and Szymanski [17] use statistical analysis and machine learning to find node traits and activities that correlate well with the formation and persistence of links and can predict social network evolution. Nowell and Kleinberg [18] develop approaches to link prediction based on measures of proximity of nodes in a network. Experiments on large co-authorship networks suggest that information about future interactions can be extracted from network topology alone [18]. Influential nodes when seeded (activated intentionally) may activate a large portion of the network through a viral contagion process. Goldenberg and Sela [19] suggested and analyzed a scheduled seeding approach for influence maximization.

C. Social network analysis in software development

Toral et al [20] analysed the networks of open source software projects using social network analysis methodologies. They developed macro structural and micro structural analyses. The macro structural analysis identified the communities responsible for the efficient development of the project. The micro structural analysis identified brokerage as the key role to be performed by the communities. Sowe et al [21] show that knowledge brokers are important people in open source software projects and are expert human resources. Teixeira et al [22] applied social network analysis to explore the role of groups, sub communities and business models in the Openstack ecosystem [1].

D. Exponential random graph models

Jackson has elaborated the concept of Exponential Random Graph Models (ERGMs) in his book *Social and Economic Networks* [4]. Pol et al. [23] mentions the increasing use of ERGMs in the social networks because of their ability to explain the global structure of a network while allowing inference on tie prediction on a micro level.

In this study we considered two different software development ecosystems and examined four different networks. Existing network models such as the Erdos Renyi model and preferential attachment model provide frameworks to analyse random graphs and networks [24]. However, their main limitation is that they do not fully represent the complexities of real world networks where nodes represent human being. Hence, we used an ERGM based approach which can capture a wide range of network tendencies by using structural elements from the network. The formulation and application of the ERGMs for social networks have been elaborated by Robins et al. [25] and the techniques for approximating a maximum likelihood estimator for an ERGM given a network data have been presented in [26] and [27].

[1] https://www.openstack.org/

E. Applications of ERGM

Ghafouri and Khasteh [28] show a number of applications of ERGM and also review its applications in the study of scientific collaboration networks. Liang et al. [29] applied ERGMs to the generation of social networks in the artificial society, and a general process of generating social networks is proposed. ERGMs have also been used to understand longitudinal engagement, performance and social connectivity [30].

Jiao, Wang et al. [31] have used ERGM to analyze the character of peer relationship networks and their effects on the subjective well-being of adolescents. Yon et al. [32] have shown how ERGMs can be applied to small networks also.

Hunter, Goodreau and Handcock [33] evaluate new procedures to find how well the model fits the observed graph. Morris, Handcock and Hunter [34] describe the means for controlling the Markov Chain Monte Carlo (MCMC) algorithm used for estimation. ERGMs have been applied in recommender systems as discussed in [35]. Degree only models did a poor job of capturing an observed network structure [36]. ERGMs have also been used to examine the structures in large social networks [36], [37].

Belkhiria et al. [38] show an interesting application of ERGMs in determining nomadic herders' movement. In multilevel network contexts, ERGMs offer a statistical framework that captures complicated multilevel structure through some simple structural signatures or network configurations based on tie dependence assumptions. Wang et al. [39] review the multilevel network data structure and multilevel ERGM specifications and show that within level nodal attributes can affect multilevel network structures.

III. Study Setting

A. Overview

We examined data from the Openstack and Eclipse development ecosystems as shared by Gonzalez-Barahona et al. [40]. Openstack is an open source cloud computing platform that is available as infrastructure-as-a-service (IaaS) [2]. Eclipse is a plug-in based integrated development environment that is extensively used for software development in many languages [3]. The data shared by Gonzalez-Barahona et al. have information on developer interactions across multiple different but related development activities – source code change, problem ticket resolution, code reviews, and developer communication via mailing lists [40]. The Openstack dataset has four databases: source code (135 repositories; 183,413 commits; 3,836 authors), problem tickets (55,044 tickets; 635,895 updates; 7,582 identities), mailing list (15 lists; 88,842 messages; 4,399 posters), and reviews (119,989 code reviews; 3,533 submitters). The Eclipse dataset also has four databases: source code (492 repositories; 987,671 commits; 3,753 authors), problem tickets (470,397 tickets; 3,380,817 updates; 51,629 identities), mailing list (253 lists; 386,034 messages; 19,642 posters), and reviews (37,460 code reviews; 1,033 submitters).

TABLE I
Characteristics of the Networks

Network	Nodes	Links	Density	Diameter
Openstack Comments	826	9390	0.02749223	6
Openstack Changes	853	17019	0.04672584	5
Eclipse Comments	412	4808	0.05678785	7
Eclipse Changes	420	6467	0.07349699	7

B. Pre-processing and filtering data

Developers are involved in various activities in a large scale software development ecosystem. The most engaged developers are the ones who contribute to multiple such activities. For each dataset we identified common developers who participated in *all* four activities: resolving problem tickets, reviewing code, committing code changes, and posting messages in the mailing lists with a view to identifying the most active developers. This filtering strategy enables us to identify those developers who are most deeply embedded in the ecosystem and are thus likely to display diverse interaction characteristics. Having identified such developers, we constructed *co-comment networks* and *co-change networks* for both Openstack and Eclipse using the construction protocols we describe next.

C. Construction of networks

For each of the following network types, nodes represent the common developers who participated in all four development activities (as discussed above) and two nodes are connected by a link if the developers corresponding to the nodes at either end of the link have both participated in some common unit of development activity. In a *co-comment network*, two developers are connected by a link if both of them have co-commented on a particular problem ticket. In a *co-change network*, two developers are connected by a link if both of them have co-changed a particular unit of code. For both these types of networks, the weight of a link signifies the number of co-commenting or co-changing instances between the developers connected by that link. For each networks we removed the singleton nodes as they do not have links incident on them.

The structural parameters of our networks (which we will refer to as Openstack Comments, Openstack Changes, Eclipse Comments, Eclipse Changes) are presented in Table I For each network, we calculated the following six metrics for each node: degree centrality, closeness centrality, betweenness centrality, eigenvector centrality, pagerank and clustering coefficient. We next discuss why we chose these particular metrics in this study.

D. Choice of network metrics

Degree centrality has always been a highly effective measure of the influence or importance of a node. We generally see that in social settings, people with more connections tend

[2] https://en.wikipedia.org/wiki/OpenStack

[3] https://en.wikipedia.org/wiki/Eclipse_(software)

to have more power and are more visible. However, we posit that this might not be the case in every situation. Given the richness and variety of interactions in a software development ecosystem, other characteristics of a node can be important indicators of developer interaction.

Betweenness centrality measures the number of times a node lies on the shortest path between other nodes. By the definition of betweenness, it is an important identifier for individuals with notable influence; high betweenness of a node signifies it is in a position of brokerage between other nodes.

Closeness centrality scores each node based on their closeness to all other nodes in the network. This measure calculates the shortest paths between all nodes, then assigns each node a score based on the sum of its shortest paths. This means that individuals with high closeness are the ones who are best placed to influence the entire network quickly. Closeness can be regarded as a measure of how long it will take to spread information from a node to all other nodes.

Similar to degree centrality, eigenvector centrality measures a node's influence based on the number of links it has to other nodes in the network. Eigenvector centrality also takes into account factors such as how well connected a node is, and how many links their connections have. By calculating the extended connections of a node, eigenvector centrality can identify nodes with influence over the whole network, and not just those directly connected to it.

Pagerank is a variant of eigenvector centrality, also assigning nodes a score based on their connections, and their connections' connections. The difference is that pagerank also takes link direction and weight into account so links can only pass influence in one direction, and pass different amounts of influence. Since it takes into account direction and connection weight, pagerank can be helpful for understanding citation dynamics and authority.

Clustering coefficient is a measure of the degree to which nodes in a network tend to cluster together. Evidence suggests that in most real-world networks, and in particular social networks, nodes tend to create tightly knit groups characterized by a relatively high density of ties; this likelihood tends to be greater than the average probability of a tie randomly established between two nodes. In Table II we summarize the specification and formulation of the metrics, and outline their relevance in our study setting.

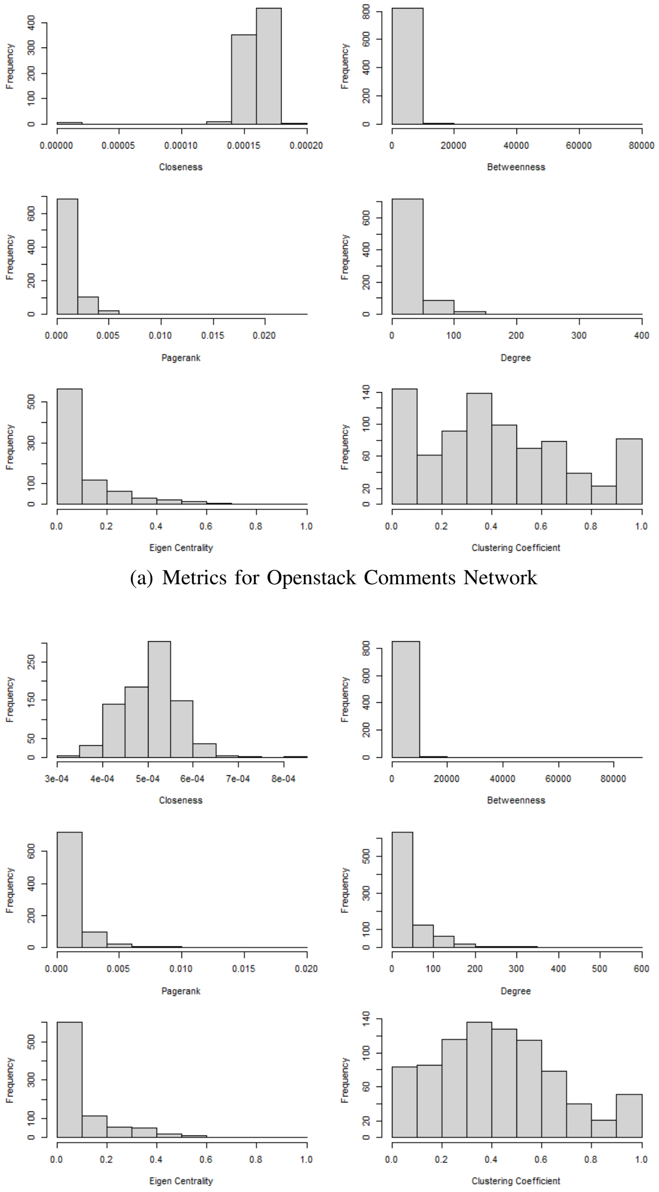
The distributions of these metrics for each of our networks are presented in Figures 1 and 2 and their descriptive statistics are shown in the Tables III, IV, V, VI.

IV. Methodology

An overview of the research methodology is presented in Figure 3. We motivate our approach using a simple example.

A. Calculating network metrics

Figure 4 shows a simple network with five nodes (labelled as "1", "2", "3", "4", and "5") and six links. Using the formulations presented in Table II, the metrics for each of the five nodes node of this network are calculated and presented in Table VII:



(a) Metrics for Openstack Comments Network

(b) Metrics for Openstack Changes Network

Fig. 1. Openstack Distributions

B. Application of ERGMs

As indicated earlier, Exponential Random Graph Models take into consideration the local features and dependencies of nodes. With this, we can estimate the statistics of the network and understand the parameters that drive network formation. As we have emphasized before, every node has its own attributes which need to be considered. Our observed network is one instance of a large number of alternative networks which may or may not have similar features. An ERGM is a statistical model that considers all the alternative networks and then provides inference on the factors influencing the formation of a particular network's structure. Additionally, ERGM predicts the probability that a pair of nodes in a network will have a

TABLE II
AN OVERVIEW OF THE METRICS

Metric	**Specification**	**Formulation**	**Relevance for this study**
Degree centrality	The number of links incident upon a node.	Count the number of links a node has [3].	Degree indicates the total number of connections a developer has with his/her peers and signifies how many communication channels the developer participates in.
Closeness centrality	Reciprocal of the sum of the length of the shortest paths between the node and all other nodes in the network.	$C(x) = \frac{1}{\sum_y d(y,x)}$ [3]	Closeness is a measure of how effectively information spreads from a node; a developer who is close to many other developers will be favourably positioned to exchange information.
Betweenness centrality	Number of times a node lies in the shortest path between two other nodes in the network.	Take every pair of nodes in the network and count how many times a particular node lies in the shortest paths (geodesic distance) between the two nodes of the pair [3].	Betweenness is an indicator of the extent to which a node can broker communication between other nodes; a developer with high betweenness can act as bridge between diverse groups of peers who are otherwise unconnected.
Eigen centrality	This is calculated by scoring the relative importance of all nodes in the network by weighting connections to highly important nodes more than connections to nodes of lower importance.	$x_v = \frac{1}{\lambda} \sum_{t \epsilon M(v)} x_t$ where x_v is the eigen centrality of node v, $M(v)$ is the set of neighbours of v and λ is a constant [3].	Eigen centrality identifies nodes with influence over the whole network; developers with high eigen centrality are members of the project team who are strongly positioned to influence their peers.
Pagerank	Computed by counting the number and quality of links to a Web page to determine how important the page is, in a network representing the World Wide Web. The underlying assumption is that more important Web pages are more likely to receive more links from other pages.	$PR(u) = \sum_{v \epsilon B_u} \frac{PR(v)}{L(v)}$ the pagerank value for a page u is dependent on the pagerank values for each page v contained in the set B_u (the set containing all pages linking to page u), divided by $L(v)$, which the number of links from page v [3].	Pagerank is an indicator of importance and authority of a node in a network; developers with high pagerank are in positions of enhanced expertise and experience relative to their peers.
Clustering coefficient	Measure of the extent to which a node is clustered together with other nodes in the network.	$C_i = \frac{n_t}{n_r}$ where n_t is the number of triangles connected to node i and n_r is the number of triples centred around node i. A triple centred around node i is a set of two links connected to node i [3].	Clustering coefficient measures the extent to which a node belongs to clusters; a developer with high clustering coefficient participates to a larger extent in collaborative activities with his/her peers.

TABLE III
STATISTICS FOR OPENSTACK COMMENTS NETWORK METRICS

Metrics	Mean	Median	Std Dev
Closeness	1.586558×10^{-4}	1.608105×10^{-4}	1.430113×10^{-5}
Betwenness	6.604104×10^{2}	27.36651	3.595547×10^{3}
Degree	22.73608	10	33.1497
Pagerank	1.210654×10^{-3}	7.230619×10^{-4}	1.596366×10^{-03}
Eigen Centrality	0.1090396	0.05177937	0.1456735
Clustering Coefficient	0.4224369	0.3900663	0.2959932

TABLE IV
STATISTICS FOR OPENSTACK CHANGES NETWORK METRICS

Metrics	Mean	Median	Std Dev
Closeness	5.05001×10^{-4}	5.09165×10^{-4}	6.107212×10^{-5}
Betwenness	5.792251×10^{2}	30.99205	3.204844×10^{3}
Degree	39.90387	20	53.22984
Pagerank	1.172333×10^{-3}	7.141349×10^{-4}	1.40447×10^{-03}
Eigen Centrality	0.09949386	0.04495683	0.1313177
Clustering Coefficient	0.4274744	0.4065041	0.252135

TABLE V
STATISTICS FOR ECLIPSE COMMENTS NETWORK METRICS

Metrics	Mean	Median	Std Dev
Closeness	9.82674×10^{-4}	9.775171×10^{-4}	1.649528×10^{-4}
Betwenness	3.185558×10^{2}	20.15124	7.511532×10^{2}
Degree	23.33981	11	29.18822
Pagerank	2.427184×10^{-3}	1.443223×10^{-3}	2.442064×10^{-03}
Eigen Centrality	0.180408	0.08444197	0.2231443
Clustering Coefficient	0.5658584	0.5509039	0.2708142

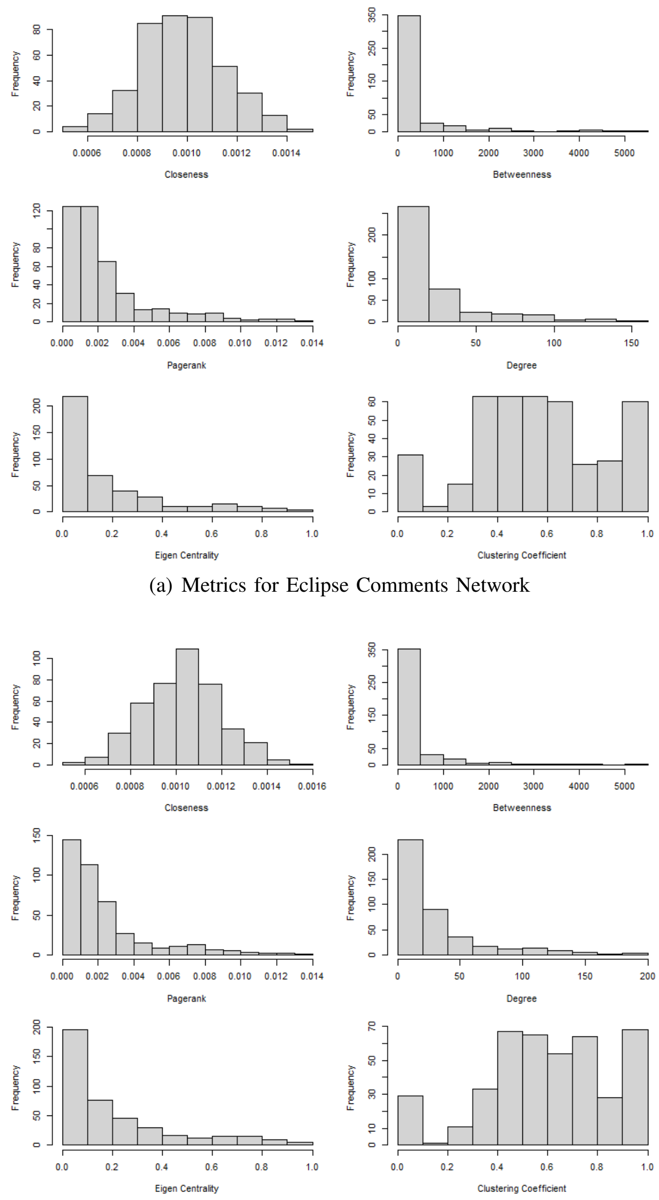
TABLE VI
STATISTICS FOR ECLIPSE CHANGES NETWORK METRICS

Metrics	Mean	Median	Std Dev
Closeness	1.028976×10^{-3}	1.040583×10^{-3}	1.695752×10^{-4}
Betwenness	2.906714×10^{2}	18.28643	6.749869×10^{2}
Degree	30.79524	17	37.60238
Pagerank	2.380952×10^{-3}	1.556057×10^{-3}	2.398271×10^{-03}
Eigen Centrality	0.2091423	0.118087	0.2378225
Clustering Coefficient	0.6085609	0.6088854	0.2642028

link between them on the basis of node attributes [23]. Thus in our context ERGM allows us to identify developer attributes that influence interactions between developers.

As discussed, ERGM is a generative model that takes into consideration local properties and dependencies of the nodes. The probability of a network formation is given by the following equation:

$$Pr(g) = \frac{exp(\beta S(g))}{\sum_{g'} exp(\beta S(g'))} \tag{1}$$



(a) Metrics for Eclipse Comments Network

(b) Metrics for Eclipse Changes Network

Fig. 2. Eclipse Distributions

TABLE VII
METRICS OF THE SIMPLE NETWORK' VERTICES

Metrics	Vertex 1	Vertex 2	Vertex 3	Vertex 4	Vertex 5
Degree	3	2	3	2	2
Closeness	0.083	0.071	0.083	0.058	0.058
Betweenness	2	0	2	0	0
Eigen centrality	1.000	0.650	1.000	, 0.951	0.951
Pagerank	0.230	0.142	0.230	0.198	0.198
Clustering coefficient	0.333	1.000	0.333	0.000	0.000

The network that we feed to the ERGM is denoted by g. $S(g)$ is the statistic of the network g. We can use different statistics such as the number of links, number of triangles etc.; β is the coefficient that determines how important the statistic would be for g. g' denotes all possible networks with the given number of nodes. Here, $Pr(g)$ determines how likely a network is formed due to situations that are influential in a particular context, and not merely due to happenstance.

We now return to our simple illustrative example (see Figure 4) for the network with five nodes (n = 5) and six links (e = 6) to demonstrate how ERGMs function. Let us consider one network statistic: the number of links. Now, the maximum number of links possible in a network with n nodes is ${}^{n}C_2$, which in this case is ${}^{5}C_2 = 10$. The number of networks possible with 5 nodes is $2^{{}^{n}C_2}$ which is $2^{10} = 1024$ in this case. However, we actually have 6 links in our network. Hence, the frequency of link formation will be $6/10 = 0.6 = pr$ (say).

As explained by Jackson [4] $\beta = log(\frac{pr}{1-pr}) = log(\frac{0.6}{0.4}) = 0.176$. An ERGM estimates the parameter that best suits the network using Markov Chain Monte Carlo (MCMC) estimation. When we fit the ERGM to this network [4], the estimate is computed as 0.4055. While interpreting this estimate, we find out the log odds of a link occurring which is equal to $0.4055*$ *change in the number of links* = $0.4055 * 1$, as the addition of any link to a network changes the number of links in the network by 1. The corresponding probability is $exp(0.4055)/(1 + exp(0.4055)) = 0.6$ [4]. This is exactly what we calculated above as $6/10 = 0.6 = pr$, since there are $6/10$ links in our network. In the context of this example, a simple illustration of how ERGM functions is shown in the Figure 5.

In this study we take each of the four empirical networks – Openstack Comments, Openstack Changes, Eclipse Comments, Eclipse Changes – and use ERGMs to estimate the influence of the specific network metrics identified in Table II in the formation of links in these networks. We next discuss our results.

V. RESULTS AND DISCUSSION

A. Model parameters

The results from fitting the ERGMs to the respective networks are presented in Tables VIII, IX, X, XI. In these tables, the significance levels indicated in superscripts of the parameter estimates, as determined by the respective p-values are as follows: 0 is denoted by $***$; 0.001 is denoted by $**$; 0.01 is denoted by $*$; 0.05 is denoted by . ; 0.1 is denoted by blank space. The parameter estimates of the model give the log odds of a tie occurring. They give us the probability of the network with respect to our assumed predictors. This helps us understand whether the network occurs due to the influence of the predictors or just by chance[5]. The higher the magnitude of the estimate, the more influence the predictor has on the response. A positive value implies that there is

[4] http://cran.nexr.com/web/packages/ergm/vignettes/ergm.pdf

[5] https://cran.r-project.org/web/packages/ergm/vignettes/ergm.pdf

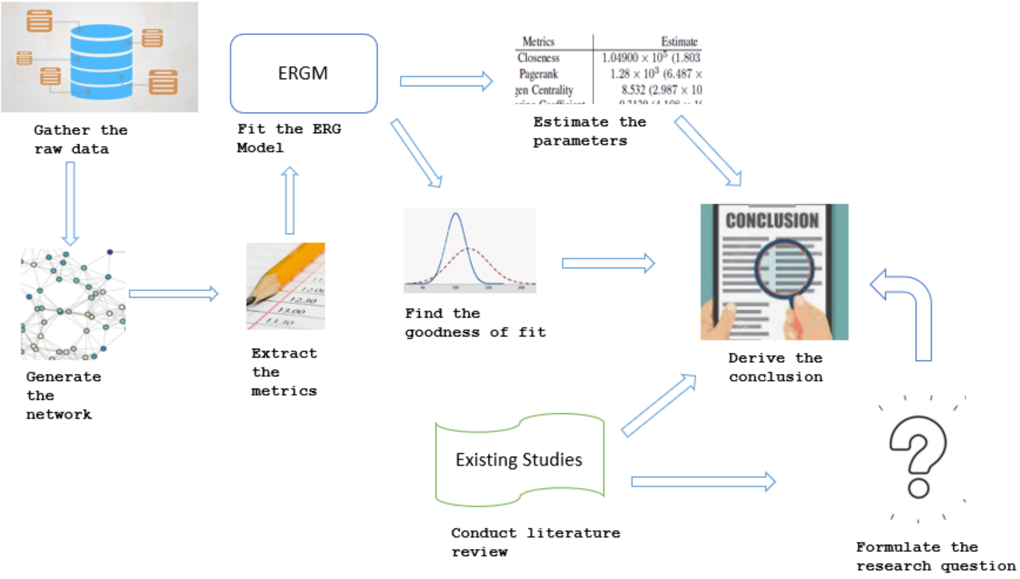


Fig. 3. Research methodology

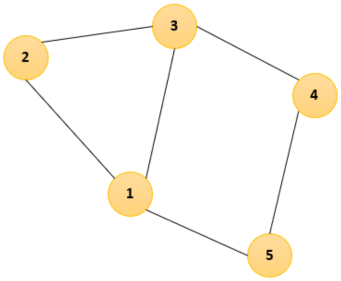


Fig. 4. A simple example

TABLE VIII
PARAMETER ESTIMATES FOR OPENSTACK COMMENTS NETWORK

Metrics	Estimate
Closeness	8.47×10^4 (1.939×10^3) ***
Clustering Coefficient	0.1292 (4.388×10^{-2}) **
Degree	0.05163 (2.623×10^{-3}) ***
Betweenness	-2.679×10^{-5} (2.658×10^{-6}) ***
Eigen Centrality	-1.129 (2.563×10^{-1}) ***
Pagerank	-6.428×10^2 (3.663×10^1) ***

positive predictor effect on link formation, while a negative value changes direction of the effect [26].

B. Evaluating model fit

In order to check how well our model fits the data, we analysed the goodness of fit of the ERGMs for all four networks. As mentioned earlier, ERGMs are generative models; that is, they seek to abstract the processes that govern link formation at a local level. These local processes aggregate to produce global network properties, even though the global properties are not explicit terms in the model. Whether a model fits the data can thus be checked by examining how well the model

TABLE IX
PARAMETER ESTIMATES FOR OPENSTACK CHANGES NETWORK

Metrics	Estimate
Closeness	1.04900×10^4 (1.803×10^{02}) ***
Pagerank	1.28×10^3 (6.487×10^{01}) ***
Eigen Centrality	8.532 (2.987×10^{-1}) ***
Clustering Coefficient	0.3139 (4.108×10^{-2}) ***
Betweenness	-6.26×10^{-5} (2.388×10^{-6}) ***
Degree	-0.03977 (2.177×10^{-3}) ***

TABLE X
PARAMETER ESTIMATES FOR ECLIPSE COMMENTS NETWORK

Metrics	Estimate
Closeness	7.217×10^3 (2.854×10^2) ***
Pagerank	7.888×10^2 (6.731×10^1) ***
Clustering coefficient	0.1505 (9.924×10^{-2})
Betweenness	-5.319×10^{-4} (3.864×10^{-5}) ***
Degree	-0.03977 (5.722×10^{-3}) ***
Eigen Centrality	-0.7373 (3.843×10^{-1}) .

TABLE XI
PARAMETER ESTIMATES FOR ECLIPSE CHANGES NETWORK

Metrics	Estimate
Closeness	8.051×10^3 (3.068×10^2) ***
Pagerank	9.218×10^2 (8.553×10^1) ***
Eigen centrality	0.591 (3.871×10^{-1})
Betweenness	-5.701×10^{-4} (5.141×10^{-5}) ***
Degree	-0.04757 (4.846×10^{-3}) ***
Clustering coefficient	-0.2 (1.008×10^{-1}) *

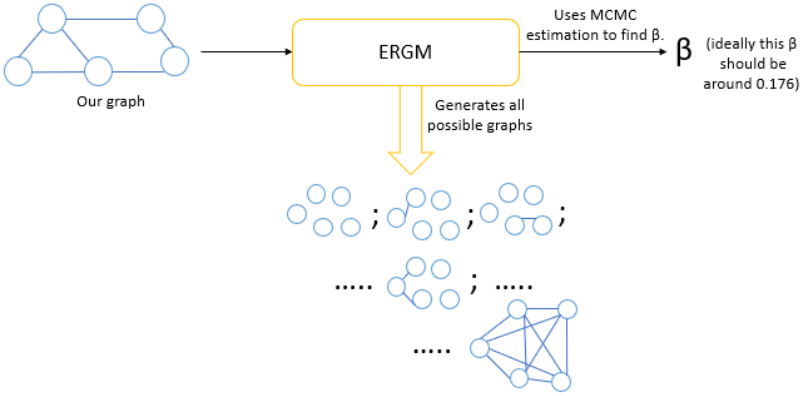

Our graph
ERGM
Uses MCMC estimation to find β.
β
(ideally this β should be around 0.176)
Generates all possible graphs
;
;
;
.....
;
.....
.....

Fig. 5. ERGM Illustration

reproduces the global properties of the network [4]. We can do this by choosing a network statistic that is not in the model, and comparing the value of this statistic as observed in the empirical networks, to the distribution of the statistic's values from networks simulated by our model. For this purpose, we chose the *minimum geodesic distance* as this is an important global characteristic of a network; but is not one of the metrics included in our model. The smallest number of links that connect two nodes is the geodesic distance between them and the path is called the geodesic path [6].

Figure 6 shows how the empirical value of the minimum geodesic distance for each network compares with the value from the corresponding model. The black line in the figures represent the observed statistic of the empirical networks and the box plots show the mean statistics from the corresponding models. The closer the observed statistics are to the mean statistics from the models, the better the model fits the corresponding data. We observe from Figure 6 that for higher values of the minimum geodesic distance, the empirical and model generated values are notably close for all four networks, indicating a reasonably good model fit.

C. Observations

In conventional wisdom, degree of a node in a network is considered to be a pre-eminent indicator of the node's position in the ecosystem that the network represents [41], [42]. Thus it is natural that degree is expected to be influential in the formation of connections between the entities that the nodes represent. As is evident from Tables VIII, IX, X, XI, magnitudes of the parameter estimates for closeness centrality are the highest among across all four networks, and all such estimates are statistically significant. *Thus among all the metrics we have considered, closeness has the maximum influence on interconnection between developers. Contrary to expectations, we do not find evidence that degree is the key driver of link formation in the networks we studied.* After closeness, pagerank has the next highest influence on link formation in three out of our four networks. Evidently, there is a clear indication that closeness and pagerank are the most important factors in determining developer interactions in our study setting.

D. Implications

As outlined in Section III, closeness of a node indicates how close it is to other nodes in the network, as measured in terms of the shortest paths between the nodes. In a software development ecosystem, developers with higher closeness are positioned to connect with other developers with minimal communication overheads. Thus the *interaction cost* for such developers is relatively low in comparison with other developers. The impact of such costs in large scale software development have long been recognized [43]. Our results underscore the need to focus on closeness rather than degree when assessing developers' interaction needs such that communication costs can be optimized.

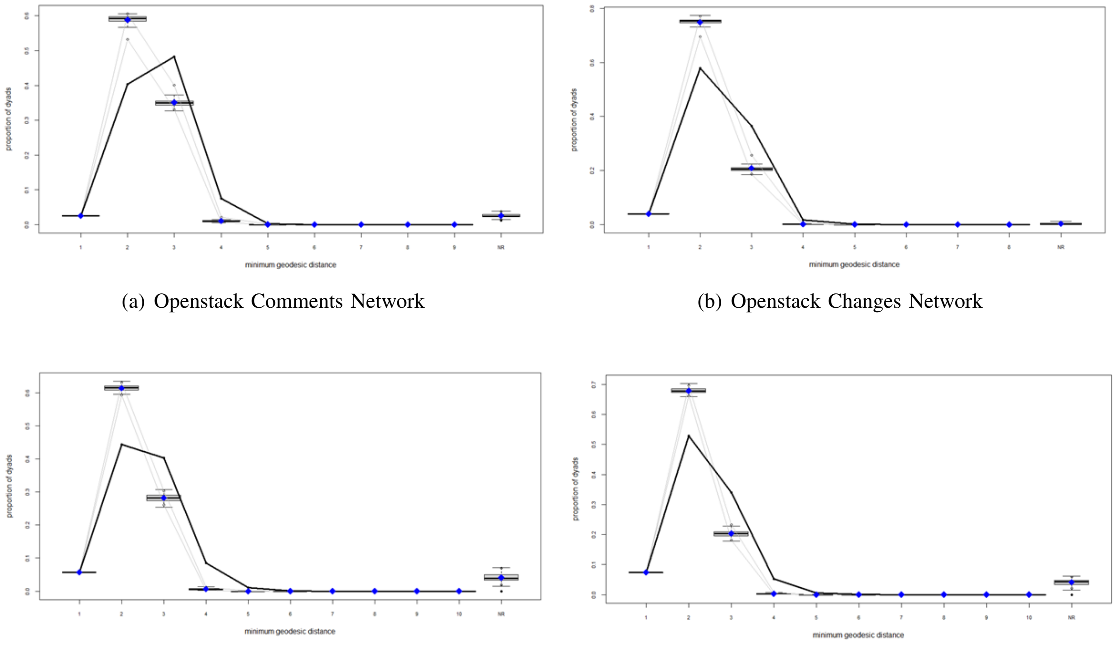
We find evidence that after closeness, pagerank is a notable determinant of interactions between developers. Pagerank is a widely recognized measure of importance of a node in a network. Thus our results indicate that it is more likely for important developers to connect to one another. In a software development ecosystem, importance of a developer can be indicative of enhanced experience, expertise, and concomitant organizational seniority. The pre-eminence of developer importance as indicated by pagerank – vis-a-vis developer connectivity as indicated by degree – in influencing link formation highlights that channels of information flow in a software development ecosystem are predominantly between those who are more strongly positioned to share such information.

Our findings have notable utility at individual, team, and organizational levels, which we will discuss next.

E. Utility

The evidence that closeness followed by pagerank are the most important determinants of interactions in large scale software development ecosystems, can inform **individual developers** on the most effective ways to collaborate. Instead of

[6] https://www.sci.unich.it/~francesc/teaching/network/geodesic.html



(a) Openstack Comments Network

(b) Openstack Changes Network

(c) Eclipse Comments Network

(d) Eclipse Changes Network

Fig. 6. Goodness of fit: Minimum geodesic distance

indiscriminately connecting with many of their peers – thus leading to high degree – developers will be better served if they cultivate higher levels of closeness with other important developers in the project. This practice will enable individual developers to effectively leverage the “tribal memory” of a project ecosystem – the combined collection of skills, know-how, and techniques that accumulates in the developer community over time [44].

Our results challenge some of the existing mores of **project management**. The reliance on degree as the primary parameter reflecting on developers’ extent of interconnectedness, is questioned by the evidence we see of the importance of closeness, followed by pagerank. For a particular developer, being merely connected to many other developers is found to be less consequential than being close to those who occupy a position of importance in the project milieu. This insight can help project managers enable each developer to cultivate a close circle of peers, as well as culture skills and expertise in focussed areas of interest.

For **organizations** engaged in the delivery of software solutions via geographically distributed teams of developers, team assembly and governance present enduring challenges [45]. Interaction patterns that are organic in co-located teams are difficult to replicate in distributed teams. Such teams whose members are spatially and temporally segregated face distinct challenges in their outcomes [46]. Connection, separation, association, and clustering among developers have contrasting effects on the quality of teams’ work products [47]. Our results illuminate the nuanced nature of developer interaction in large and distributed teams. Among all the network metrics we considered that reflect on the nature and extent of developer communication, closeness and pagerank are found to be dominant drivers of how developers connect with one another. These findings can contribute to key decision making processes in organizations when development work for particular software modules need to be allocated to teams with developers separated by distance and time-zones. The evidence that for a particular developer, closeness to other developers and importance amongst his/her peers plays a critical role in how the developer will connect to other developers, can be a valuable input into the mechanisms by which organizations assign individuals to teams.

VI. Threats To Validity and Future Work

A. *Threats to validity*

We present results from an observational study. We will now identify and address threats to the validity of our results in terms of *construct validity, internal validity, external validity* and *reliability* and outline plans of future work.

1) Construct validity: Construct validity is concerned with the extent to which our variables are measured correctly. Our variables are network metrics that are calculated using established procedures. For all the four networks, we have calculated every metric by the standard measure as shown in Table II. Exponential Random Graph Models are now being used extensively in the study of social networks, as discussed in preceding sections. However we understand that other recent modelling approaches like the Subgraph Generation Models

(SUGM) or hybrid models might also offer interesting insights in our study setting. Different construction protocols for the networks can also lead to different results.

2) Internal validity: Internal validity is established if a study is free from systematic errors and biases. We studied historical data from the Openstack and Eclipse development ecosystems as curated and published for research purposes [40]; thus issues such as mortality and maturation do not pose threats in our study setting. We have considered two types of developer interaction networks for each of the two development ecosystems we studied; replicating our study on other types of networks can offer an opportunity to widen the insights from our results.

3) External validity: External validity is concerned with the generalizability of a study's results.We have studied two software development ecosystems and have drawn insights from them. Thus, we do not claim our results to be generalizable as yet. Further studies with other ecosystems may lead to additional insights.

4) Reliability: Reliability relates to the extent to which the results from a study can be reproduced. Our results are fully reproducible. To facilitate the replication of our results, we have shared the code components developed for this project at https://github.com/IshitaB28/R.

B. Plans of future work

In this paper we present empirical evidence that closeness and pagerank are among the influential drivers of link formation in networks of developers. In our future work, we plan to expand our study setting to include additional development ecosystems, as well as other types of development activities such as code review. Also, we plan to explore the causal mechanisms that underlie the results from our statistical models to address questions around why closeness and pagerank – rather than degree and the other network metrics we considered – predominantly drive link formation between developers in our study setting.

VII. Summary and Conclusions

An in-depth understanding of the drivers of developer communication in large scale software development ecosystems has strong implications at individual, project, and organizational levels. In this paper we have examined a research question around identifying developer attribute that maximally influence developer interaction in such ecosystems. Using data from multiple development activities in two large real-world software development ecosystems, we construct networks whose nodes represent developers and two developers are linked if they co-participate in some particular development activity. Analysis of these networks using exponential random graph models (ERGMs) offer evidence that closeness and pagerank are the two most important node properties influencing link formation in these networks. From these results we infer that contrary to conventional wisdom, degree is not the key determinant of whether and how individuals connect with one another in our study setting. Instead, the extent to which developers are close to other developers and the importance of developers among their peers play pre-eminent roles in determining developer interaction. These results can inform individual developers on the most effective interaction practices in a project ecosystem; project managers can use these results to facilitate developer communication that is most optimal for team outcomes, and organizations can use the insights from our study in team assembly and governance.

References

[1] R. Guimera, B. Uzzi, J. Spiro, and L. A. N. Amaral, "Team assembly mechanisms determine collaboration network structure and team performance," *Science (New York, N.Y.)*, vol. 308, no. 5722, pp. 697–702, Apr. 2005, PMID: 15860629. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/15860629

[2] S. Datta, R. Roychoudhuri, and S. Majumder, "Understanding the relation between repeat developer interactions and bug resolution times in large open source ecosystems: A multisystem study," *Journal of Software: Evolution and Process*, vol. 33, no. 4, p. e2317, 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/smr.2317

[3] M. E. J. Newman, "The structure and function of complex networks," *cond-mat/0303516*, Mar. 2003, SIAM Review 45, 167-256 (2003). [Online]. Available: http://arxiv.org/abs/cond-mat/0303516

[4] M. O. Jackson, *Social and Economic Networks*. Princeton University Press, 2008. [Online]. Available: https://www.jstor.org/stable/j.ctvcm4gh1

[5] R. Albert and A. Barabasi, "Statistical mechanics of complex networks," *cond-mat/0106096*, Jun. 2001, reviews of Modern Physics 74, 47 (2002). [Online]. Available: http://arxiv.org/abs/cond-mat/0106096

[6] E. Ravasz and A.-L. Barabasi, "Hierarchical Organization in Complex Networks," *Physical Review E*, vol. 67, no. 2, p. 026112, Feb. 2003, arXiv: cond-mat/0206130. [Online]. Available: http://arxiv.org/abs/cond-mat/0206130

[7] S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes, "Pseudofractal Scale-free Web," *Physical Review E*, vol. 65, no. 6, p. 066122, Jun. 2002, arXiv: cond-mat/0112143. [Online]. Available: http://arxiv.org/abs/cond-mat/0112143

[8] F. Comellas, G. Fertin, and A. Raspaud, "Recursive graphs with small-world scale-free properties," *Physical Review E*, vol. 69, no. 3, p. 037104, Mar. 2004, arXiv: cond-mat/0402033. [Online]. Available: http://arxiv.org/abs/cond-mat/0402033

[9] L. Chen, F. Comellas, and Z. Zhang, "Self-similar planar graphs as models for complex networks," *arXiv:0806.1258 [cond-mat, physics:physics]*, Jun. 2008, arXiv: 0806.1258. [Online]. Available: http://arxiv.org/abs/0806.1258

[10] C. Song, S. Havlin, and H. A. Makse, "Self-similarity of complex networks," *Nature*, vol. 433, no. 7024, pp. 392–395, Jan. 2005, arXiv: cond-mat/0503078. [Online]. Available: http://arxiv.org/abs/cond-mat/0503078

[11] G. Golnari and Z.-L. Zhang, "Multivariate Heavy Tails in Complex Networks," in *Combinatorial Optimization and Applications*, ser. Lecture Notes in Computer Science, Z. Zhang, L. Wu, W. Xu, and D.-Z. Du, Eds. Cham: Springer International Publishing, 2014, pp. 557–570.

[12] M. O. Jackson and B. W. Rogers, "Meeting Strangers and Friends of Friends: How Random Are Social Networks?" *American Economic Review*, vol. 97, no. 3, pp. 890–915, May 2007. [Online]. Available: https://pubs.aeaweb.org/doi/10.1257/aer.97.3.890

[13] V.-A. Nguyen, C. W.-K. Leung, and E.-P. Lim, "Modeling Link Formation Behaviors in Dynamic Social Networks," in *Social Computing, Behavioral-Cultural Modeling and Prediction*, ser. Lecture Notes in Computer Science, J. Salerno, S. J. Yang, D. Nau, and S.-K. Chai, Eds. Berlin, Heidelberg: Springer, 2011, pp. 349–357.

[14] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins, "Microscopic evolution of social networks," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '08. New York, NY, USA: Association for Computing Machinery, Aug. 2008, pp. 462–470. [Online]. Available: https://doi.org/10.1145/1401890.1401948

[15] J. Leskovec, J. Kleinberg, and C. Faloutsos, “Graphs over time: densification laws, shrinking diameters and possible explanations,” in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, ser. KDD ’05. New York, NY, USA: Association for Computing Machinery, Aug. 2005, pp. 177–187. [Online]. Available: https://doi.org/10.1145/1081870.1081893

[16] C. W.-k. Leung, E.-P. Lim, D. Lo, and J. Weng, “Mining interesting link formation rules in social networks,” in *Proceedings of the 19th ACM international conference on Information and knowledge management*, ser. CIKM ’10. New York, NY, USA: Association for Computing Machinery, Oct. 2010, pp. 209–218. [Online]. Available: https://doi.org/10.1145/1871437.1871468

[17] A. Bahulkar, B. K. Szymanski, O. Lizardo, Y. Dong, Y. Yang, and N. V. Chawla, “Analysis of link formation, persistence and dissolution in NetSense data,” in *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, Aug. 2016, pp. 1197–1204.

[18] D. Liben-Nowell and J. Kleinberg, “The link prediction problem for social networks,” in *Proceedings of the twelfth international conference on Information and knowledge management*, ser. CIKM ’03. New York, NY, USA: Association for Computing Machinery, Nov. 2003, pp. 556–559. [Online]. Available: https://doi.org/10.1145/956863.956972

[19] D. Goldenberg, A. Sela, and E. Shmueli, “Timing Matters: Influence Maximization in Social Networks Through Scheduled Seeding,” *IEEE Transactions on Computational Social Systems*, vol. 5, no. 3, pp. 621–638, Sep. 2018, conference Name: IEEE Transactions on Computational Social Systems.

[20] S. L. Toral, M. R. Martnez-Torres, and F. Barrero, “Analysis of virtual communities supporting OSS projects using social network analysis,” *Information and Software Technology*, vol. 52, no. 3, pp. 296–303, Mar. 2010. [Online]. Available: https://doi.org/10.1016/j.infsof.2009.10.007

[21] S. Sowe, I. Stamelos, and L. Angelis, “Abstract Identifying knowledge brokers that yield software engineering knowledge in OSS projects,” 2006.

[22] J. Teixeira, G. Robles, and J. M. Gonzlez-Barahona, “Lessons learned from applying social network analysis on an industrial Free/Libre/Open Source Software ecosystem,” *Journal of Internet Services and Applications*, vol. 6, no. 1, p. 14, Jul. 2015. [Online]. Available: https://doi.org/10.1186/s13174-015-0028-2

[23] J. van der Pol, “Introduction to Network Modeling Using Exponential Random Graph Models (ERGM): Theory and an Application Using R-Project,” *Computational Economics*, vol. 54, Oct. 2019.

[24] L. Bernick, “Modeling Human Networks Using Random Graphs,” p. 13.

[25] G. Robins, P. Pattison, Y. Kalish, and D. Lusher, “An introduction to exponential random graph (p*) models for social networks,” *Social Networks*, vol. 29, no. 2, pp. 173–191, May 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378873306000372

[26] D. R. Hunter, M. S. Handcock, C. T. Butts, S. M. Goodreau, and M. Morris, “ergm: A Package to Fit, Simulate and Diagnose Exponential-Family Models for Networks,” *Journal of statistical software*, vol. 24, no. 3, p. nihpa54860, May 2008. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2743438/

[27] M. Handcock, D. Hunter, C. Butts, S. Goodreau, and M. Morris, “Statnet: Software Tools for the Representation, Visualization, Analysis and Simulation of Network Data,” *Journal of statistical software*, vol. 24, pp. 1548–7660, Feb. 2008.

[28] S. Ghafouri and S. H. Khasteh, “A survey on exponential random graph models: an application perspective,” *PeerJ Computer Science*, vol. 6, p. e269, Apr. 2020, publisher: PeerJ Inc. [Online]. Available: https://peerj.com/articles/cs-269

[29] L. Liang, G. Yuanzheng, and Q. XiaoGang, “Using exponential random graph - models to generate social networks in artificial society,” in *Proceedings of 2013 IEEE International Conference on Service Operations and Logistics, and Informatics*, Jul. 2013, pp. 596–601.

[30] M. Zhu, Y. Bergner, Y. Zhang, R. Baker, Y. Wang, and L. Paquette, “Longitudinal engagement, performance, and social connectivity: a MOOC case study using exponential random graph models,” in *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, ser. LAK ’16. New York, NY, USA: Association for Computing Machinery, Apr. 2016, pp. 223–230. [Online]. Available: https://doi.org/10.1145/2883851.2883934

[31] C. Jiao, T. Wang, J. Liu, H. Wu, F. Cui, and X. Peng, “Using Exponential Random Graph Models to Analyze the Character of Peer Relationship Networks and Their Effects on the Subjective Well-being of Adolescents,” *Frontiers in Psychology*, vol. 8, 2017, publisher: Frontiers. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fpsyg.2017.00583/full

[32] G. G. Vega Yon, A. Slaughter, and K. de la Haye, “Exponential random graph models for little networks,” *Social Networks*, vol. 64, pp. 225–238, Jan. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378873320300496

[33] D. R. Hunter, S. M. Goodreau, and M. S. Handcock, “Goodness of Fit of Social Network Models,” *Journal of the American Statistical Association*, vol. 103, no. 481, pp. 248–258, Mar. 2008. [Online]. Available: https://www.tandfonline.com/doi/full/10.1198/016214507000000446

[34] M. Morris, M. S. Handcock, and D. R. Hunter, “Specification of Exponential-Family Random Graph Models: Terms and Computational Aspects,” *Journal of statistical software*, vol. 24, no. 4, pp. 1548–7660, 2008. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2481518/

[35] D. H. Yang and Y. Su, “A Social Recommender System Based on Exponential Random Graph Model and Sentiment Similarity,” *Applied Mechanics and Materials*, vol. 488-489, pp. 1326–1330, 2014, conference Name: Materials Science, Civil Engineering and Architecture Science, Mechanical Engineering and Manufacturing Technology ISBN: 9783037859766 Publisher: Trans Tech Publications Ltd. [Online]. Available: https://www.scientific.net/AMM.488-489.1326

[36] S. M. Goodreau, “Advances in exponential random graph (p*) models applied to a large social network,” *Social Networks*, vol. 29, no. 2, pp. 231–248, May 2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378873306000402

[37] C. J. Anderson, S. Wasserman, and B. Crouch, “A p* primer: logit models for social networks,” *Social Networks*, vol. 21, no. 1, pp. 37–66, Jan. 1999. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378873398000124

[38] J. Belkhiria, M. Lo, F. Sow, B. Martnez-Lpez, and V. Chevalier, “Application of Exponential Random Graph Models to Determine Nomadic Herders’ Movements in Senegal.” *Transboundary and Emerging Diseases*, vol. 66, Apr. 2019.

[39] P. Wang, G. Robins, and P. Matous, “Multilevel Network Analysis Using ERGM and Its Extension,” Dec. 2016, pp. 125–143.

[40] J. M. Gonzalez-Barahona, G. Robles, and D. Izquierdo-Cortazar, “The metricsgrimoire database collection,” in *Proceedings of the 12th Working Conference on Mining Software Repositories*, ser. MSR ’15. Piscataway, NJ, USA: IEEE Press, 2015, pp. 478–481. [Online]. Available: http://dl.acm.org/citation.cfm?id=2820518.2820592

[41] S. Milgram, “The small-world problem,” *Psychology Today*, vol. 1, pp. 61–67, May 1967.

[42] D. J. Watts, *Six Degrees: The Science of a Connected Age*, 1st ed. W. W. Norton & Company, Feb. 2003.

[43] F. P. Brooks, *The Mythical Man-Month: Essays on Software Engineering, 20th Anniversary Edition*. Addison-Wesley, 1995.

[44] G. Booch, “Tribal memory,” *IEEE Software*, vol. 25, no. 2, pp. 16–17, 2008.

[45] V. Gilrane, “Working together when we are not together,” https://www.blog.google/inside-google/working-google/working-together-when-were-not-together/ Last accessed: April 11, 2019, 2019.

[46] P. Wagstrom and S. Datta, “Does latitude hurt while longitude kills? geographical and temporal separation in a large scale software development project,” in *Proceedings of the 36th International Conference on Software Engineering*, ser. ICSE 2014. New York, NY, USA: Association for Computing Machinery, May 2014, pp. 199–210. [Online]. Available: https://doi.org/10.1145/2568225.2568279

[47] S. Datta, “How does developer interaction relate to software quality? an examination of product development data,” *Empirical Software Engineering*, vol. 23, no. 3, pp. 1153–1187, Jun. 2018. [Online]. Available: https://doi.org/10.1007/s10664-017-9534-0