3-2016

Collaboration in scientific digital ecosystems: A socio-technical network analysis

Philip Mutuma Munyua
Purdue University

Graduate School Form 30
Updated 1/15/2015

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By PHILIP MUTUMA MUNYUA

Entitled
COLLABORATION IN SCIENTIFIC DIGITAL ECOSYSTEMS: A SOCIO-TECHNICAL NETWORK ANALYSIS

For the degree of Doctor of Philosophy

Is approved by the final examining committee:

SABINE BRUNSWICKER
Chair

THOMAS J. HACKER

SORIN A. MATEI

KATHYRNE A. NEWTON

To the best of my knowledge and as understood by the student in the Thesis/Dissertation Agreement, Publication Delay, and Certification Disclaimer (Graduate School Form 32), this thesis/dissertation adheres to the provisions of Purdue University's "Policy of Integrity in Research" and the use of copyright material.

Approved by Major Professor(s): SABINE BRUNSWICKER

Approved by: KATHYNE NEWTON 3/7/2016

Head of the Departmental Graduate Program Date

COLLABORATION IN SCIENTIFIC DIGITAL ECOSYSTEMS: A SOCIO-TECHNICAL NETWORK ANALYSIS

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Philip Mutuma Munyua

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

May 2016

Purdue University

West Lafayette, Indiana

To my late mom: you've gone too soon…

ACKNOWLEDGEMENTS

I would like to thank my committee members, Dr. Sabine Brunswicker (Chair), Dr. Kathryne A. Newton, Dr. Sorin A. Matei and Dr. Thomas J. Hacker.

I would like to thank my family; Dad, Jason Munyua. My late mom. Jennifer Munyua, Siblings, Millicent Makena, John Kithinji, Catherine Kathambi. Nieces, Doreen Nana, Ashley Makandi, Florida Nkatha, Natalie. To all of whom, your support and encouragement has been warm and the much needed impetus in the entire process.

I would also like to thank my research team at Research Center for Open Digital Innovation and nanoHUB.org (Network for Computational nanotechnology) for great team work in solving many research and coding issues. I would particularly like to acknowledge Kang-Yu, Kerina, Cherly, Lynn, Michael and Dwight. You are an amazing team. There are so many people that I would like to thank personally: People that have been involved in one way or another (directly and indirectly) in the entire study process.

The NanoHub Network Analysis Project, partially supported by the KredibleNet project (an NSF funded project-Award 1244708), created the dataset used in this dissertation. My work for this dissertation was partially funded by the Exploratory Research in Social

Sciences Grant (PI: Sorin Adam Matei) and by Research Center for Open Digital Innovation (PI: Sabine Brunswicker) to whom I acknowledge.

TABLE OF CONTENTS

Page

Page

Page

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Table Page

ABSTRACT

Munyua, Philip Mutuma. Ph.D., Purdue University, May 2016. Collaboration in Scientific Digital Ecosystems: A Socio-Technical Network Analysis. Major Professor: Sabine Brunswicker.

This dissertation seeks to understand the formation, operation, organizational (collaboration) and the effect of scientific digital ecosystems that connect several online community networks in a single platform. The formation, mechanism and processes of online networks that influence members output is limited and contradictory. The dissertation is comprised of three papers that are guided by the following research questions: How does online community member's productivity (or success) depend upon their 'position' in the digital networks? What are the network formation mechanism, structures and characteristics of an online community? How do scientific innovations traverse (diffuse) amongst users in online communities? A combination of exploratory, inductive and deductive research designs is applied sequentially but in a non-linear manner to address research question. The dissertation contributes to the literature on scientific collaboration, digital communities of creation, social network modelling and diffusion of innovation.

The first paper applies network theory and spatial probit autocorrelative modelling technique to evaluate how member developer's positioning in digital community correlate with his/her productivity. The second paper looks at the dynamics

of developer's participation in online developers' network for a period spanning 7-years using exponential random graph models (ERGM). This paper applies theory of network (network science) to model network formation patterns in developer community. The third paper, like the first, applies network theory and to understand user network characteristics and communication channels which influence diffusion of scientific innovations. Bass and spatial probit autocorrelative models are applied for this analysis.

Data from this study was mined from developers, authors and user communities of nanoHUB.org cyberinfrastructure platform. NanoHUB.org is a science and engineering online ecosystem comprising self-organized researchers, educators, and professional communities in eight member institutions that collaborate, share resources and solve nanotechnology related problems including development and usage of tools (scientific innovation). Data from collaboration and information sharing activities was used to create the developers, authors and user networks that were used for analysis.

Results of the first paper show that the spatial autocorrelation parameter of the spatial probit model is negative and statistically different from zero. The negative spatial spillover effect in the developer network imply that developers that are embedded in the network have a lower probability of getting more output. The structural network characteristics of eigen vector centrality had statistically significant effects on probability of being more productive. Developers who are also authors were found to be more productive than those in one network. The implications of these findings is that developers will benefit from being in multiple network spaces and by associating with more accomplished developers. The autocorrelative and interaction models also reveal

various new modelling approach of accounting for network autocorrelation effects to online member.

Results of the second paper show that developers form in a manner that follow a pure uniform random distribution. Results also show that developer's collaborative mechanisms are characterized by low tendencies to reciprocate and form homophiles (tendency of developers to associate with similar peers) but high tendency to form clusters. The implications of network formation mechanism and processes are that developers are forming in a purely random and self-organized manner and minimum efforts should be applied in trying to organize and influence the community organization. The results also reveal that a simple link to link ERGM and stochastic dominance criteria can be combined to characterize the network formation characteristics just like the ERG(p*) model but have an advantage of overcoming degeneracy challenges associated with ERG(p*) models.

Results of the third paper show that bass model is a good predictor for diffusion of scientific innovations (tools) in online community setting. Results also show different innovations have varying levels and rates of adoption and these were influenced by both external and internal factors. Results of the micro-based model found degrees and betweeness centrality as some of the internal variables that have positive influence on the adoption of innovation while centrality measures of power or leadership were found to have negative influence of adoption process. The relative time taken to run a simulation (measured as job usage time) was also found to be negatively influencing diffusion. The implication of the study results is that bass model is a good fit for evaluating and forecasting adoption of innovation in online communities. Moreover, network structural

characteristics are responsible for adoption of innovation adoption and policy making should consider tool adoption enhancing ones. Additionally, researchers could further explore the network structural characteristics that are driving diffusion of innovation.

CHAPTER 1. INTRODUCTION

1.1 Introduction

The global systems of scientific collaboration and communication have been changing and growing rapidly in the last two decades due to improvements in information and communication technologies (Brunswicker et al., 2015; Schroeder, Jennifer, deBeer & Fry, 2007). The growth has transformed customary collaboration practices of innovation and production including the "traditional" research and collaboration practices in various field of science (Schroeder, Jennifer, deBeer & Fry, 2007). The traditional collaboration[1] and systems of digital practice were mostly enabled by three channels (formal, informal and tabular) and primary, secondary (library catalogs and indexing services) and tertiary (encyclopedias and reviews) sources (Sondergaard, Anderson & Hjorland, 2003). The three channels and sources were first singled out in 1971 by the United Nations Educational scientific and Cultural organizations (UNESCO) and International Council of Scientific Unions (ICSU), UNISIST model as mechanisms which enabled member scientist to collaborate (Sondergaard et al., 2003). Gold (2007) and Faraj and Johnson (2011) noted that the changes and ongoing growth in systems of scientific communication have also affected both formal and informal communication and data

[1] Communication and collaboration are used interchangeably throughout this study to imply engagements.

sharing and dissemination methods, "gatekeeping"[2] and outputs. The transformations in scientific systems of communication and collaboration have elicited research interest about the new form of scientific organization because collaborations and communication in those platforms is voluntary and the collaboration mechanics are self-organizing (Brunswicker et al., 2015; Faraj & Johnson; 2011; Levine & Prietula, 2014; Matei, 2014; Matei et al., 2015). Research interest in these platforms has focused on "why", "how" and "what", that is, why do participants enter, how does the platform maintain itself (and in most cases grow) and what do members gain by being in those platforms. Faraj and Johnson (2011) noted that online-based platforms are characterized by large networks of people/scientists that would not have been possible without communication that is highly efficient (e.g., high speed internet). Other factors that have been known to influence growth of online platforms include allowing access through mainly open collaboration model and availing resources like data and simulation tools[3] to participating members (Gold, 2007; Levine and Prietula, 2014; nanoHUB.org, 2014). There is, therefore, a worldwide effort to make scientific research on collaboration and communication and practice a permanent part of scientific data research through platforms where processing, storage and dissemination of data through open 'access' model (open source) is gaining popularity (Faraj and Johnson, 2011; Gold, 2007). In this study we will focus on the changes (growth) and organization of such kind of online platform known as nanoHUB.org cyberinfrastructure (e.g., Brunswicker et al., 2015; Matei, 2014). We

[2] "Gatekeeping is the process through which information is filtered for dissemination, whether for publication, broadcasting, the Internet, or some other mode of communication" (Barzilai, 2009)
[3] Tools are scientific artifacts (softwares) used to run simulations and applications programs including data visualization (nanoHUB.org, 2014)

particularly focused on emerging data, information sharing method and outputs using a platform called NanoHub.org[4] cyberinfrastructure (A detailed description of the platform is discussed under research design)

1.1.1 The Emergence of Scientific Cyberinfrastructures and Digital Communities

Scientific cyberinfrastructure was initially used by the US National Science Foundation (NSF) in early 2000 to denote broad and unified systems of software, hardware, middleware and networks that are designed to better manage big data; procurement, mining, storage, amalgamation and visualization over the internet. i.e., a computer technology based infrastructure for information and communication (Gold, 2007; Stewart et al., 2010). Cyberinfrastructure is also known by the terms e-science and e-infrastructure in UK (United Kingdom) and EU (European Union) respectively (Schroeder et al., 2007). The scientific community defines cyberinfrastructure as, "…infrastructure consisting of computational systems, data and information management, advanced instruments, visualization environments, and people[5], all linked together by software and advanced networks to improve scholarly productivity and enable knowledge breakthroughs and discoveries not otherwise possible" (Stewart et al., 2010). Gold (2007) found that research in cyberinfrastructure involves evaluation of

[4] NanoHUB.Org is a scientific cyberinfrastructure (ecosystem) that involves scientific tool developers, tool users, authors, educators and learners that work in a novel self-organizing and distributed way to produce, use, and learn with scientific software tools (Brunswicker et al., 2015; https://nanohub.org/)

[5] Scientist, actors, developers and users are used interchangeably throughout this study to imply online community members

computing systems, data storage structures, data repositories and innovative instruments, graphical settings, and scientists (people) that are all interconnected by high speed internet to make possible scholarly innovation and discoveries that would have otherwise not been possible. Kling, McKim and King (2003) established that social structures formed by scientists in their organizations are needed in addition to advancement in technology and communication. The authors further noted that social structures provide an informal system of social and technical (socio-technical) interaction which facilitates scholarly scientific communication. i.e., communication is driven by technology but it is also defined by the social structures of participating scientists and their groups. As early as 1980's, Abelson (1980) had also described scientists as inherently "social" and usually connected through formal or informal collaboration in communications that enable scientific progress. As it will be seen below, cyberinfrastructure platforms mostly facilitate scientist's collaboration through allowing scientists to interact at will thorough an open collaboration model (Levine & Prietula, 2014).

Open collaboration is a model that allows the general public to freely access a source code for their use and/ or also for modification from its original plan (Levine & Prietula, 2014). Several techniques for managing and allowing access to the source code exist including what Levine and Prietula (2014) described as the harbinger for open collaboration; open source. The most generally known open source is open source software (OSS). Crowston, Wei, Howison & Wiggin (2012) described OSS as "a software which is released under a license that permits inspection, use, modification, and redistribution of the software's source code by volunteer programmers". The volunteer programmers come together virtually and form OSS communities while working on the

software. Some commonly known OSS examples include the Linux operating system and the Apache Web Server-http (the largest and most successful OSS), user applications (e.g., Mozilla Firefox, OpenOffice), Internet infrastructure (e.g., sendmail, bind) and programming language interpreters and compilers (e.g., Python, gcc) (Crowston et al., 2012).

The OSS community has been growing tremendously since the inception of the OSS model in the late 1990 (Ursula, 2004). Vass (2007) estimated that OSS community has 800,000 programmers/scientists around the world and the number continues to grow making OSS an important portion of the collaboration infrastructure of modern digital society. The growth in OSS and OSS community has seen an equal increase in the bulk of studies examining the digital open collaboration occurrence (e.g., Crowston et al., 2012; Rossi, 2006). The majority of this literature is comprised of studies that have modelled OSS communities as network spaces involving actors who form and break ties (collaborate) in that space based on their inherent goals (e.g Abbasi, Chung, & Hossain, 2012; Brunswicker et al., 2015; Gonzalez-Brambila, Veloso, & Krackhardt, 2013; Matei, 2014). In this study I will follow Matei (2004) and Brunswicker et al. (2015) social network and spatial autocorrelation perspective to model online collaboraties as networks that form, grow and contribute to members' outcomes (e.g., Abbasi et al., 2012; Borgatti & Halgin, 2012; Gonzalez-Brambrila et al., 2013; Jackson, 2008)

1.1.2 Statement of the Problem

The growth in technology (cyberinfrastructure) enabled online communities has made the platforms a vital part of the collaboration infrastructure of the current society because the networks formed in the online communities are seen as sources or facilitators

of information that is relevant to member's productivity. Technology based online communities are distinguished by a unique and novel form of organization that is characterized by members that join the platforms voluntarily and has those members self-organize themselves and maintain (or grow) the networks. This new form of organization has drawn researchers into examining the networks from several aspects including, one, effect of networks on participant's outcomes (productivity[6] or choices), and two, patterns of formation and sustenance mechanisms in technology enabled online communities (e.g., Abbasi et al., 2012; Brunswicker et al., 2015; Crowston et al., 2012; Faraj & Johnson, 2011; Gonzalez-Brambila et al., 2013; Jackson & Rogers, 2007; Matei, 2004; Rossi, 2006; Scacchi, 2007).

A large proportion of literature is comprised of studies that have modelled online platforms as network spaces involving actors who form and break ties (collaborate) in that space based on their inherent goals (e.g., Jarvenpaa & Leidner, 1999; Kanawattanachai & Yoo, 2007; Kankanhalli, Tan & Wei, 2005; Wasko & Faraj, 2000&2005). Others have looked at the effect of the networks on members' outcome when measured as productivity or choice (e.g., Abbasi et al., 2012; Brass, 2002; Brunswicker et al., 2015; Gonzalez-Brambila et al., 2013; Jackson and Rogers, 2007). Borgatti and Halgin (2011) and Matei (2014) established that success is usually measured using social capital while choice is usually measured as an aspect of social homogeneity caused by contagion processes.

[6] Productivity is measured at the number of citations a developer receives through citations of developed tools.

The mechanisms and processes of collaboration which influence output and diffusion processes in the established network is limited, nevertheless. There are very few studies of scientific production that have looked at the interactions and characteristics of network structures as factors of production despite its importance in understanding the collaboration mechanisms (e.g., Abbasi et al., 2011; Brunswicker et al, 2015; Gonzalez-Bambrila et al., 2013; Li et al., 2013; McFadyen & Cannella, 2004; Matei, 2014; Singh, 2007;). There are even lesser studies that have looked at these interactions and characteristics using network autocorrelation model that would best capture the global effects of those networks on member's success (e.g., Brunswicker et al., 2015; Matei, 2014) and none, to our knowledge, that has looked at the interactions in multiple (two or more) networks in a digital infrastructure/platform. On the diffusion processes, there are few empirical studies that have looked at the effect of network on diffusion (e.g., Ballester, Calvo-Armengol & Zenou, 2006; Banerjee, Chandrasekhar, Duflo, & Jackson, 2013; Meade & Islam, 2006), and no study has looked at diffusion from a network autocorrelation perspective in a non-market based digital platform.

Moreover, the above highlighted network effect techniques only describe and understand the network characteristics and their effects on community member's productivity and choice; they rarely address the network formation and sustenance mechanism which is also not well understood (Faraj & Johnson, 2011; Jackson & Rogers, 2007; Matei, 2014; Robins, Pattison, Kalish, & Lusher, 2007). This study therefore seeks to fill the above highlighted three knowledge gaps and is guided by the following research questions.

1.1.3 Research Questions

This study seeks to understand formation, operation, organizational (collaboration) and the effect of networks formed in online digital communities to members and is guided by the following research questions,

1. How does productivity (measured as number of citations a developed tool gets) of members of online communities depend upon their positioning in the digital networks?
2. What are the network formation and sustenance mechanism and structural characteristics of a digital platform?
3. How do innovations traverse (diffuse) amongst user network in online digital platforms?

The research questions are addressed in form of three independent papers that combine socio-technical tools. The first paper broadly applies network theory and spatial econometrics technique to evaluate how developer's positioning (embeddedness) in digital space correlate with his/her output. The second paper looks at the network formation and sustenance mechanism and structural characteristics of developer network. This paper broadly applies theory of network (network science) to model patterns in network formation and sustenance mechanism. The third paper, like the first, broadly applies network theory and spatial econometrics to understand user network characteristics that influence diffusion of scientific tools. The motivation, model specification and results of the three papers are discussed in details below.

1.1.4 Theoretical Foundation

This study is anchored on network analysis primarily concerned with evaluating the effect and formation mechanism of networks in digital (online) platforms following Brunswicker et al. (2015) and Matei (2014) study of evolution of digital practice capital. Digital platforms enable members to form digital practice enabled networks through source coding, tool usage and other computer enabled associations and engagements that are mostly facilitated by the platform's API (Application Program Interface) (Brunswicker et al., 2015; Matei, 2014; nanoHUB.org, 2014). In network analysis, the study will broadly focus on network theory and theory of networks (Borgatti & Halgin, 2011). While noting that analysis and definition of the two theories is subtle, Borgatti and Halgin (2011) defined network theory as the study of the outcome associated with mechanisms and processes that occur within a network structure and theory of network as the study that determines why network form. i.e., models of which scientist/actors form ties (links, triads, e.t.c) and how they position themselves (e.g., centrality measures, small-worldness e.t.c) the network as a whole will have. Network theory asks questions like what will be the effect of network structural characteristics like having high degree centrality (many ties) or betweeness centrality (being centrally located (e.g., Brass, 2002). Theories (sub-theories) that have emerged from network theory includes the well-known strength of weak ties by Granovetter's (1973) and Burt's (1992) structural holes (SH). Borgatti and Halgin (2011) noted that these theories that have been used widely to study network features on outcomes and are usually tested by network coordination model or the network flow model.

The network coordination model is based on the structure and position of scientists in a network (Borgatti & Halgin, 2011). For example, a weak tie will be valuable in SWT models because they link network clusters/components. i.e., their position in the network (structural role) makes them valuable in that setting (Burt, 1992). In SH, the shape of the ego network (personal network/1-neighborhood/first-order zone,) around a scientist/actor gives them advantage to others that are positioned in other clusters. Therefore, network structures and attributes interactions are examined through either choice (social homogeneity) or success (social capital) outcome[7] variables where, for example, one could explore the effects of network structural differences on any of the two variables (Borgatti & Halgin, 2011).

The network flow model is also called the implicit theory of network function (Borgatti & Halgin, 2011). The authors noted that this model assumes that SWT and SH sub-theories depend on a basic model of a social systems that form networks that facilitate information to flow. Some theoretical propositions derived from this model would be influenced by SH and SWT theories and would include network measures such as distance (location of the nodes which determines time of information arrival) and embeddedness (this determines the relevancy of information received i.e., on-redundant flow received) (Borgatti & Halgin, 2011; Jackson & Rogers, 2007). Furthermore these network measures are then correlated to more common outcomes that have traditionally

[7] Network theory models are often used to explain two broad type of outcomes: one, the choice outcome i.e., behavioral, attitudes, beliefs and internal structural characteristics as for the case of organizations, and, two, the success outcome which includes parameters like performance and or rewards (Borgatti & Halgin, 2011)

been evaluated using either of the two outcomes (Borgatti & Halgin, 2011; Jackson & Rogers, 2007). Table 1 shows a schematic representation of the tradition research in network analysis.

Table 1: Traditional Research in Network Analysis

	Research Tradition	
Model	Social Capital	Social Homogeneity
Network flow model (ties as pipes)	Capitalization	Contagion
Network coordination (ties as bonds)	Cooperation	Convergence

Source: Borgatti & Halgin, 2011.

The columns in the Table 1 shows the two traditional areas of research in social networks, social capital and social homogeneity while the rows show the network models (measures). Research work in contagion includes diffusion models or adoption models (Borgatti & Halgin, 2011; Jackson, 2008). These models test networks as flow models (i.e., ties as pipes) where, for example, information symmetry is reached through information flow (conduit) in the network. Borgatti and Halgin (2011) noted that research on convergence includes evaluating networks as bonds that, say coordinate information or resources to some converging measures e.g., research on structural equivalence while research on capitalization has mostly tested the concept of social capital theory in SWT and SH. The authors further noted that cooperation research consist of bond-based explanations of achievements.

Research of theory of network has mostly involved evaluation of the network formation processes as either random (e.g., Erdos Renyi, ERGM-Exponential Random

Graph Models), preferential or hybrid model involving both processes (Jackson, 2008; Lusher et al., 2013). These models evaluate the network from the scientist's behavioral point of view i.e., by looking at models of which scientist/actors form ties (links, triads, e.t.c) and how do they position themselves (e.g., centrality measures, small-worldness e.t.c) (Brass, 2002; Jackson, 2008). This study will therefore apply both theories; network theory and theory of network to answer the research questions. Network theory will be used to address the first and third research questions in papers 1 and 3 and theory of network for research question two (corresponding to paper 2).

1.1.4.1 Research Design

This study combined observation, induction, and deduction research designs in all the three papers (Recker, 2013). Exploratory Analysis is first used to understand patterns of the data. The observed patterns were then used to rationalize the data (inductive reasoning) that helped us derive some set of hypothesis. The hypotheses were then tested and validated through statistical analysis to make deductions about our rationalization/hypotheses. "Deduction is commonly used to predict the results of the hypotheses or propositions" and the validated results (deductions) were then used to prove or disapprove our hypothesis and/ or theory where applicable (Recker, 2013). The application of the three research design was done in a sequential manner but updated regularly based on the findings of predicted results. This made the process non-linear as shown in Figure 1 below.

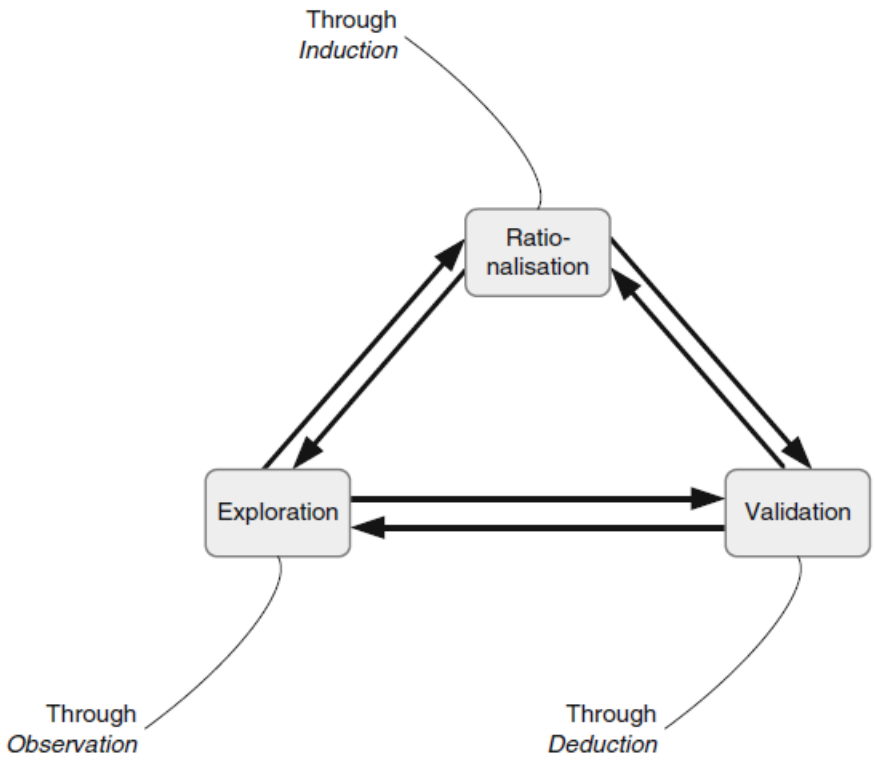


Figure 1: Exploration, rationalization and validation in research design (Source: Recker, 2013)

1.1.4.2 Study Platform/ Cyberinfrastructure

NanoHUB.org cyberinfrastructure platform was used to explore, rationalize and validate our study on emerging data, information sharing method and outputs. The platform was used to mine data from developer, author and user communities of nanoHUB.Org Cyberinfrastructure. NanoHUB.org is a science and engineering cyberinfrastructure that supports research efforts in nanoelectronics in "eight member institutions (including Purdue University, the University of California at Berkeley, the University of Illinois at Urbana-Champaign, Massachusetts Institute of Technology, The Molecular Foundry at Lawrence Berkely National Laboratory, Norfolk State University, Northwestern

University, and the University of Texas at El Paso" (Klimeck et al. 2008). The hub was created by network for computational nanotechnology (NCN) in 2002 with the support of US National Science foundation, national nanotechnology initiative. Our data from nanoHUB.org is organized by the communities of scientists that form the platform. The communities in nanoHUB.org are comprised of users from research, education, and industry who come together (form networks) to develop tools, learn from each other and use tools for their personal use or class related work, that is, run simulations e.t.c.(McLennan, 2012).

1.1.5 Problem Background and New Contribution

Digital practice has been articulated in the context of *NanoHUB.org Network Analysis Project* (Matei, 2014), to which I contributed as a research assistant and on which I build upon my research. The project was dedicated to explaining online social collaboration through social network and spatial autocorrelation lenses. The theoretical justification for using these methodological tools was proposed by Matei (2009 and 2014) and, building on this conceptualization, in Brunswicker et al. (2015). The core concept is that of social collaborative practice, an evolutionary understanding of the social capital and coordination concept (e.g. Abassi et al., 2011; Gonzalez-Brambila et al., 2013 and Li et al., 2013). The other perspective is that of social autocorrelative research developed in social sciences with an interest in spatial problem. Such research included a rich literature (e.g., Leenders, 2002; O'Malley & Marsden, 2008) but the more direct source of inspiration of our current work are Brunswicker et al (2015) and the NanoHub Social Network Analysis Project. My contribution to this research is to extend the research on digital social practice capital methodologically in three folds. One, as an extension of the

work proposed by Brunswicker et al. (2015) and Matei (2014), I explore the degree to which digital practice capital has a direct and real influence collaborative productivity. This is attained by incorporating more relational aspects of network effect models as applied by (Abassi et al., 2011; Li et al., 2013; Gonzalez-Brambila et al., 2013) and though through network autocorrelation modelling which enables us to capture the global effects of the network (Brunswicker et al., 2015; Matei, 2014). Two, I explore how digital social practice emerge and evolve. Specifically, I am interested in finding out the network structural characteristics that are responsible for the evolution of digital practice capital and coordination. Finally, and more importantly, one of the core contribution of tis dissertation, which goes beyond the models proposed previously, is to explore the degree to which digital practice capital and coordination is responsible for diffusion of innovation. My research build on the dataset produced by the NanoHub Network Analysis Project. The conceptualization of the network, especially, in terms of gravitational attraction between collaborators, was defined in the dataset and I am using it as such. The explanation I provide for the network building methodology is a recounting of the methodology pioneered by Matei (2014) in the context of studying open source collaborative processes (Matei et al., 2015)

1.1.6 Scope

The first paper characterizes network positioning/embeddedness variables that are correlated with developer’s productivity and also identifies whether being embedded in multiple network spaces is more advantageous than one. The second paper identifies the patterns of formation of developer network and also identifies the network characteristics

that sustain the growth of the network. The third paper determines the rate of diffusion of tools in the user network and also identifies user and network characteristics that enhance diffusion of tools in nanoHUB.org. A Schematic representation of the nanoHUB.org platform is presented in Figure 2.

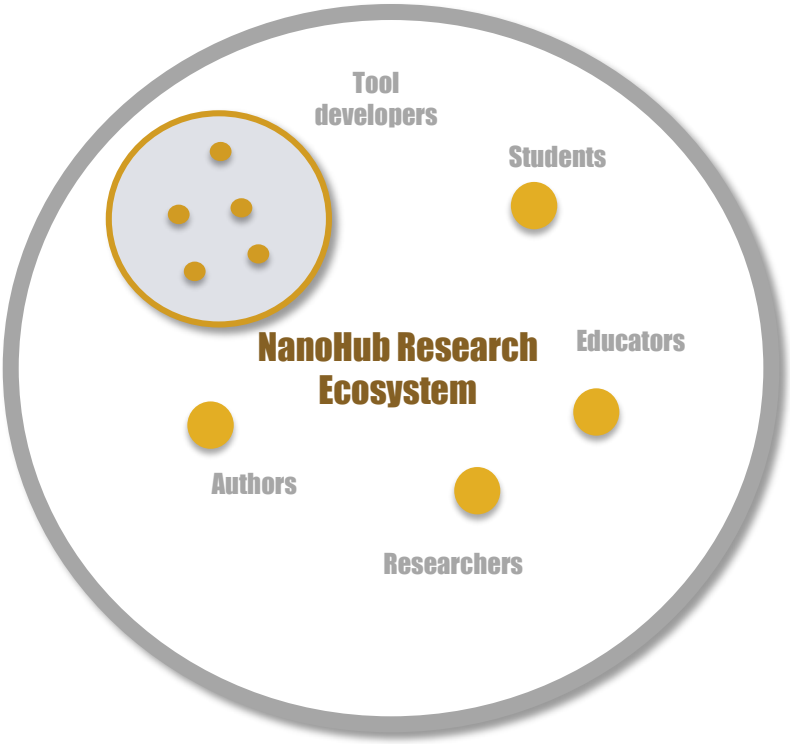


Figure 2: Schematic Representation of nanoHUB.org Platform/Ecosystem

Figure 2 shows the nanoHUB.org cyberinfrastructure platform. The platform is comprised of several network spaces that are used for this study including developer, authors and tool users (students, researchers and educators).

1.1.7 Definitions and Acronyms

1.1.7.1 Definitions

Cyberinfrastructure: "Cyberinfrastructure consists of computational systems, data and information management, advanced instruments, visualization environments, and people, all linked together by software and advanced networks to improve scholarly productivity and enable knowledge breakthroughs and discoveries not otherwise possible" (Stewart et al., 2010).

Embeddedness: This is a multidimensional variable relating generally to the importance of social networks of members benefits. i.e., embeddedness indicates that scientists who are integrated in dense clusters or multiplex relations of social networks face different sets of resources and constraints than those who are not embedded in such networks (Moody & White, 2003).

Gatekeeping: This is "the process through which information is filtered for dissemination, whether for publication, broadcasting, the Internet, or some other mode of communication" (Barzilai, 2009).

Innovation: This is the "mutation" of an institution or product which "incessantly revolutionizes" the original form of an institution or product. i.e., the process of developing a new and useful solution to the existing old one (Schumpeter, 1942).

Online Communities: An online community is a large virtual community whose members interact with each other primarily via the Internet for individual members or social welfare (Faraj & Johnson, 2011)

Open source: This is defined as, "a program in which the source code is available to the general public for use and/or modification from its original design free of charge" (Crowston et al., 2012).

Nanotechnology: This is the understanding and utilization of matter on the atomic and molecular scale (NanoHUB.org, 2014).

Platforms: Platforms are defined as either internal or external. Gawer and Cusumano (2013) defined internal platforms as "a set of assets organized in a common structure from which a company/organization can efficiently develop and produce a stream of derivative products". The author also defined external (industry) platforms as, "products, services, or technologies that are similar in some ways to the internal assets but which provide the foundation upon which outside firms (organized as a "business ecosystem") can develop their own complementary products, technologies, or services

Productivity: Productivity is defined as the effectiveness of developing quality tools that have a high probability of getting a cite (Daskovska et al., 2010)

1.1.7.2 Acronyms

GMM: Generalized Methods of Moment (GMM)

MCMC: Bayesian Monte Carlo Markov Chain (MCMC)

OSS: Open Source Software

SAR: Spatial Autoregressive Model

SDEM: Spatial Durbin Error Model

SLX: Spatially lagged explanatory variables Model.

1.2 Dissertation Outline

The first chapter, above, provided the introduction to the study. The chapter provided a background, highlighted research gaps, developed research questions and provided the theoretical background that encompasses the study. Chapter 2 contains a literature review of network science on outcomes and facilitation and network formation. The literature review focuses on two broad frameworks; one, the structural characteristics of the network that look at a network as a facilitation and production units responsible for increased output and information flow and, two, the network formation and sustenance aspects that keep the network in place and in most cases grow. Chapters 3 to 5 present independent papers that address each of the three research questions. The chapters start by motivating research, then provide some theoretical background and hypothesis to be tested. The chapters' then present the proposed methodology for testing the hypothesis present results and conclusion. Chapter 6 provides the summary of the dissertation. The chapter gives a synopsis of each study and then concludes by discussing limitations of the study and future work.

CHAPTER 2. LITERATURE REVIEW

2.1 Literature Review Outline

This literature review focuses on two broad frameworks; one, the structural characteristics of the network that looks at network as production units or facilitators of information flow and, two, the network formation and sustenance aspects that keep the network in place and in most cases grow. The literature on the structural characteristics of the network is further reviewed from networks as production units that facilitate member's productivity and networks as communication channels and social structures that facilitate diffusion of tools; the two literature review streams correspond to papers one and three respectively. This review in comprised of both the practical and theoretical aspects of the identified literature but leans more on theoretical aspect given the study design ultimate's goal of testing and validating a set of theories that are assumed to drive the network formation, sustenance and effects of member's output (Recker, 2013).

Research in open digital platforms (cyberinfrastructure) and open collaboration model of communication and collaboration has focused on "why" and "how", that is, why do participants enter and how does the platform maintain itself (and in most cases grow) and what outcomes do the platforms accord members (e.g., Barabasi & Albert, 1999; Gonzalez-Brambila et al. 2013; Faraj & Johnson, 2011; Jackson, 2008; Levine & Prietula, 2015; Matei, 2014). The growth in open collaboration model of scientific

communication and collaboration in digital platforms (cyberinfrastructure) has witnessed proliferation of studies researching the new phenomenon (e.g., Crowston et al., 2012; Rossi, 2006; Scacchi, 2007). The majority of this literature is comprised of studies that have modelled online communities as network spaces involving actors who form and break ties (collaborate) in that space based on their inherent goals. i.e., scientist collaborate to gain knowledge that will be useful to their scientific production output (e.g Abbasi et al., 2011; Brunswicker et al., 2015; Gonzalez-Brambila et al., 2013; Matei, 2014). Other studies have looked at how the networks are forming and sustaining themselves (e.g., Faraj & Johnson, 2011; Jackson & Rogers, 2007). The literature of diffusion has looked at the effect of network communication channels and social structures on adoption of innovation (e.g., Banerjee et al., 2013; Jackson, 2008). We therefore look at the three streams of literature separately below.

2.2 Literature on Networks as Production Units

The literature of networks as facilitation and production units has found networks to be positively correlated with participant's success (Gonzalez-Bambrila et al., 2013; Li et al., 2013; Rullani & Haefliger, 2013). Abbasi et al. (2011), Gonzalez-Bambrila et al. (2013), Li et al. (2013), McFadyen and Cannella (2004) and Singh (2007) applied individual-outcome models (network effect model) where networks were used to extract individual explanatory variables as inputs of scientific production. McFadyen and Cannella (2004) evaluated the relationship between network ties and scientist output and found a positive relationship. Singh (2007) also used network ties but added structural holes variables and found a positive relationship between these variables and scientific

output. However, these studies only focused on a few structural aspects of the network and largely ignored the effects of the relational dimension of the network that include different measurements of social capital e.g., relational capital, structural capital and cognitive capital (Gonzalez-Brambila et al., 2013). Relational dimension includes centrality measures such as degree, closeness, betweeness, eigen vector centrality amongst others (Abbasi et al., 2011; Gonzalez-Brambila et al., 2013; Li et al., 2013).

Abassi et al. (2011), Gonzalez-Brambila et al. (2013) and Li et al. (2013) extended Singh's (2007) study by including more aspects of social embedded characteristics including density and position in the network. Abassi et al. (2011) evaluated the co-authorship network structural aspects (including degree, closeness, betweeness and eigen vector centrality measures) on scientist scholars output (citation). The authors applied a network effect model (Poisson regression model) and found only degree, and eigen vector centrality measures had significance effect on member's output. Li et al. (2013) examined the effect of social capital embeddedness in network structure on scientist output. The authors used degree, closeness and betweeness centralities amongst other variables and found betweeness centrality to be significantly correlated with output. Gonzalez-Brambila et al. (2013) evaluated the effects of social capital relational, cognitive and structural aspects of network on scientists output and quality. The authors considered network structural characteristics (direct ties and their strength, density, structural hole, centrality and cross-disciplinary) and used extensive panel data and fixed effect models. The panel data ran from 1981 to 2002 and came from publication and citations database for scientific papers that had atleast an author from Mexico, North America. The authors found that relational aspects of scientist affected

quality while cognitive dimension affected quantity. The authors also found the structural dimension mattered to both measures of scientist outcomes; quality and quantity.

While scientific collaboration in cyberinfrastructure involves social interactions amongst scientist that are driven by a goal of producing output, there is limited understanding of the mechanisms and processes of collaboration which influence output as seen in the above literature on networks as production units (Brunswicker et al., 2015; Matei, 2014). This study aims to fill this literature gap through extension of Brunswicker et al. (2015) digital practice concept. The study looks at the interactions and characteristics of being embedded in multiple networks in a digital infrastructure by using network autocorrelation model.

2.3 Literature on Networks as Communication and Social Structures Units that Facilitate Diffusion.

The structural features of network also have influence on communication channels (information flow) that enhances diffusion of tools or innovations (Jackson, 2008). As early as in the 1980's, Rogers (1983) identified innovation, social structures that are affected by innovation (network structure), communication channel of the network and time as the main elements that facilitate diffusion. Bass model is an amassed model that describe diffusion from behavioral perspective of the entire network (Bass, 1969). Bass (1969) developed the model based on a simple premise that adopters are as innovators or imitators who interact in a user network that determines the rate and timing of adoption of innovation.

Research on diffusion of innovations has been approached from either the macroscopic (Bass Model) or microscopic (agent based) perspective or a combination of

both models (Meade & Islam, 2006; Laciana, Rovere, & Podestá, 2013). However, majority of these studies have applied simulation and analytical techniques with very little empirical evidence to buttress their findings (Ballester et al., 2006; Banerjee et al., 2013; Kitsak et al., 2010; Meade & Islam, 2006; van Eck et al., 2011;). Laciana et al. (2013) and Meade and Islam (2006) identified macroscopic research as that which considers the whole set of users while the micro considers individual users. The authors also noted that most macro-level studies have applied Bass model and are based on the assumption that users are fully connected (in a fully connected component) and are homogeneous which implies that every individual has some possibility of influencing the other through the connected network, i.e., there is social contagion due to homogeneity in the social networks. The advantage of the macro-level model is its ability to provide a simple and tractable was of looking at timing of diffusion of innovation of the population and also forecast diffusion patterns (Laciana et al., 2013; Mahajan, Muller & Bass, 1990). However, a major caveat of the macroscopic model that was pointed out by Peres, Muller, & Mahajan (2010) is its inability to provide an insight of about the processes (mostly the communication and social structures aspects of diffusion) that influence adoption, or how social interactions of actors are linked to the global social patterns like the microscopic models. Bulte and Stremersch (2004) also noted that the assumption of complete network connectedness and social contagion might not be being realistic in real world because rarely do you find fully connected individual in real world. The authors continued to note that that diffusion process (i.e., the typical logistic-S-Shaped adoption

curve[8]) does not actually come from social contagion process that is assumed in the bass model setting but due to some intrinsic tendency of heterogeneous individuals to adopt and this is better explained by microscopic models.

Microscopic models are commonly referred as agent based models because they evaluate individual behavior including the innovation characteristics, communication channels and social interactions that influence adoption (Fibich & Gibori, 2010; Laciana et al., 2013). The models relate explanatory variables (covariates) to adoption decision (Meade & Islam, 2006). The authors noted that microscopic models have the advantage of overcoming some weaknesses of the macro based models including the assumption of homogeneity of users.

This study will try to reconcile the conflicting perspectives of what drives diffusion amongst networks through an empirical application of both macro- and microscopic models. Our study will therefore contribute to the literature of understanding social structure (communication channels and social structure) aspects on information flow and diffusion of innovation.

2.4 Literature Review on Social Modelling

Literature on growth and attachment patterns (also referred to as social modelling) of online platforms has focused on network formation perspective, where actors are believed to have some preferences while attaching to other scientists in the network (e.g., Barbasi & Alfred, 1999; Faraj & Johnson, 2011; Jackson & Rogers, 2007). Earlier studies

[8] S-Shaped diffusion curve is similar to logistic function

of modelling social networks involved mostly evaluation of the network formation processes as either random (e.g., Erdos Renyi, ERGM-Exponential Random Graph Models), preferential (preferential attachment models that have distributions that are scale free –Barbasi and Alfred, 1999) or hybrid model involving both processes (Jackson, 2008). These models evaluated the network from the actor's behavioral point of view i.e., models of which scientist/actors form ties (links, triads, e.t.c) and how do they position themselves (e.g., centrality measures, small-worldness e.t.c) the network as a whole will have due to their action (Brass, 2002; Jackson, 2008). Recent network formation studies have found that actors do not follow preferential attachment while joining a group but do so randomly (e.g. Faraj & Johnson, 2011; Jackson & Roger, 2007).

Early literature of network modelling involved the mechanical processes that described the stochastic (random attachment) processes of network formation (e.g., Erdos & Renyi, 1959). The modelling has now been improved to include application of game theory tools to help understand the formation process (Jackson & Rogers, 2007). The authors also noted that social network modelling results in development of models that are either scale free networks (networks that follow a degree distribution that is power law) or uniform random networks (networks that follow negative exponential degree distribution). The first random graph model was developed by Erdos and Renyi in 1959 (Erdos & Renyi, 1959; Lusher et al., 2013). Erdos and Renyi (1959) developed a simple random graph model (uniform Bernoulli graph distribution) model that had every link having a fixed probability $p \in (0,1)$ of formation. Erdos and Renyi (1959) also assumed that formation of every link was independent of any other and the model is mostly useful for understanding certain thresholds and how networks come to exhibit certain features.

The model assumes that once the threshold is met the links will continue forming to one big component and this was identified as a major caveat of the model because this seems not to be a good representation of social networks, that is, it lacks structural effect like clustering, degree distribution and small diameter (Jackson, 2008; Lusher et al., 2013).

Improvements of Erdos Renyi (1959) model have involved relaxing the link formation independence through modifying the model to capture those important network dependency characteristics like clustering, degree distribution and small diameter (Jackson, 2008; Lusher et al., 2013). These include modelling network formation with dependencies as uniform random graph and/or by preferential treatment (e.g., Barabasi & Albert, 1999; Cooper & Frieze; 2003; Holland, 1981; Watts, 1999). Recently, hybrid models have also been developed (e.g., Jackson & Rogers, 2007; Kumar et al., 2000; Vazquez, 2003). Other extensions include stochastic block modes, exponential random graphs and newly introduced SERGMs/SUGMs (e.g., Chandrasekhar & Jackson 2012; Chatterjee & Diaconis, 2011; Frank & Strauss, 1986; Lusher et al., 2013).

Holland and Leinhardt (1981) introduced dependency in Erdos Renyi model thorough p1 models where they added within-dyads but failed to introduce other network characteristics like triads and between-dyad dependence. The P1 models failure to capture all features of network dependence together with estimation issues prompted Frank and Strauss (1986) to introduce Markov random graph models. The authors developed Markov random graph models on the basis of conditional independence amongst ties whereby, ties may spread in the network from some tie. i.e., presence of a tie may affect formation of others and hence network characteristics dependence (Lusher et al., 2013). Markov random graph models were thus able to capture the dependence of

network characteristics and therefore became accepted form of ERGM in the 1990s. The models were further popularized in social network research by Wasserman and Pattison (1996) as ERGM (p*) models. However, despite ERGM (p*) ability to capture most aspects of network, they degenerate in large data sets (Jackson, 2008; Lusher et al., 2013). This degeneracy issue is being addressed through other forms of modelling including Statistical Exponential Random Graph Models-SERGM/SUGMS which take some sample statistics from the large data set and use that for analysis of network formation mechanism (Jackson, 2008; Snijders et al., 2006). Other forms of growing social model improvements include Watt (1999) who revealed small average short distance and clustering in networks when he randomly modified links. Barabasi and Albert (1999) modelled formation of the complex World Wide Web (www) and found them to attach through preferential attachment. Albert et al. (1999) also modelled the www and found the network to exhibit small diameter. Jackson and Rogers (2007) used a simple stylized link to link model that mixed random meetings to preferential attachment on five networks and found them to exhibit different proportions of random to preferential attachment meetings. Of particular interest, the authors found Barabasi and albert (1999) complex network to also have about a third of meetings being uniformly random.

ERGM (p*) models have also been used to evaluate the network characteristics that are assumed to sustain the emerging online communities (e.g., Faraj & Johnson, 2011). Faraj and Johnson, (2011) modelled the network formation and exchange patterns in online communities using ERGM (p*) model. The authors sought to understand mostly the network sustenance and formation patterns of five online communities over a period

of 27-months. The authors noted that the communities exhibited exponential growth despite the entry being voluntary and organization being self-organizing. Our study will contribute to the literature of social modelling in online communities through application of simple link to link ERG model that has the advantage of capturing the network formation processes and also is able to identify the network structural characteristics that are responsible for the network formation.

CHAPTER 3. EMBEDDED IN MUILTIPLE NETWORK SPACES ON SCIENTIST DEVELOPMENT: HIGHER ORDER SPATIAL AND NETWORK FIXED EFFECT MODELS

3.1 Introduction

Scientific productivity has been found to be positively correlated with collaboration (e.g., Gonzalez-Bambrila., 2013; Li et al., 2013). Collaboration in science involves virtual and social interactions amongst scientist that are driven mostly by a goal of producing scientific artifacts (Brunswicker et al., 2015; Matei, 2014). Brunswicker et al. (2015), Matei (2014) and Rullani and Haefliger (2013) looked at virtual collaboration networks formed out of digital practice activities like coding as production networks that ends up playing an important role as a factor of production to the members. However, the member's positioning (or embeddedness) in those networks leads to different outcomes (e.g., Abbasi et al., 2011; Gonzalez-Brambrila et al., 2013; Li et al., 2013; Rullani & Haefliger, 2013) and the mechanisms and processes of collaboration which influence output is limited (Matei, 2014). There are very few studies of scientific production that have looked at the interactions and characteristics of network structures as factors of production despite its importance in understanding the collaboration mechanisms (e.g., Abbasi et al., 2011; Gonzalez-Bambrila et al., 2013; Li et al., 2013; McFadyen & Cannella, 2004; Singh, 2007). There are even lesser studies that have looked at these interactions and characteristics using network autocorrelation model (e.g., Brunswicker et

al., 2015; Matei, 2014) and none, to our knowledge, that has looked at the interactions in multiple (two or more) networks in a digital platform. Abbasi et al. (2011), Gonzalez-Bambrila et al. (2013), Li et al. (2013), McFadyen and Cannella (2004) and Singh (2007) applied individual-outcome models (network effect model) where scientist networks were used to extract individual explanatory variables. For example, McFadyen and Cannella (2004) evaluated the relationship between network ties and scientist output and found a positive correlation between the variables. Singh (2007) also used network ties but added structural holes variables and found a positive relationship between these variables and scientific output. Abassi et al. (2011), Gonzalez-Brambila et al. (2013) and Li et al. (2013) extended Singh's (2007) study by including more aspects of social embedded characteristics including density and position in the network. Abassi et al. (2011) evaluated the network structural aspects (including degree, closeness, betweeness and eigen vector centrality measures) on scientist scholars output (citation) and found only degree, and eigen vector centrality measures had significance effect on scientist's productivity. Li et al. (2013) examined the effect of social capital embeddedness in network structure on scientist output. The authors used, degree, closeness and betweeness centralities amongst other variables and found betweeness centrality to be significantly correlated with output. Gonzalez-Brambila et al. (2013) evaluated the effects of social capitals relational, cognitive and structural aspects of network on scientist's productivity and quality. The authors found that relational aspects of network affected quality, cognitive aspects affected quantity and structural aspects affected both quality and quantity. Brunswicker et al. (2015) evaluated the global and local impact of digital practice capital on developer's productivity using autocorrelative model. The authors

found that degree of contribution to the core of the digital practice structure and authorship capital to be positively correlated with developer's production. The authors also found the digital practice network as having negative spillover effects on developer's productivity.

Our study extends Abassi et al. (2011), Gonzalez-Brambila et al. (2013) and Li et al. (2013) individual-outcome models (network effect model) that looked at the effect of mostly local social embedded characteristics of a single network on scientists output in three ways. One, we incorporate relational aspects in the model through network autocorrelation modelling which enables us to capture the global effects of the network following Brunswicker et al. (2015). Two, we evaluate the effect of a scientist output when they are embedded in multiple networks (two or more-Here, we look at the virtual developer and authorship networks) and, three, we evaluate the scientific production in a pure digital platform. Unlike other scientific production systems, scientific production in digital ecosystem is largely dependent on the characteristics and interactions of the networks in the ecosystem because scientist rarely meet in those virtual platforms (e.g., Abassi et al., 2011; Brunswicker et al., 2015; Gonzalez-Brambila et al., 2013; Li et al., 2013; Matei, 2014).

3.2 Theoretical Background and Hypothesis

This study is founded on network analysis because we are primarily concerned with evaluating the effect of networks formed in digital (online) platforms. Digital platforms enable members to form networks through digital practice activities such as source coding and other computer enabled associations and engagements that are mostly

facilitated by the platform's API (Application Program Interface) and SVN (software versioning systems) information management files (nanoHUB.org, 2014). Brunswicker et al. (2015), Matei (2014) and Orlikowski (2000) found that networks evolve out of coding activities that the scientist engage in (digital practices) in the platforms. While looking at the effects of the networks of member's outcomes we will broadly look at both the individual local network effects and the global network effects that we hypothesize are driving productivity.

3.2.1 Network Global Effects on Developers Productivity

Online digital platforms like nanoHUB.org cyberinfrastructure serves as a platform that enable scientists to collaborate. For example, tool developers collaborate by working on a particular tool while authors collaborate when working on a particular paper in the nanoHub.org cyberinfrastructure. The work on the tools development involves digital practice activities including modification, deletion or addition of the contents while the work on papers involves both formal and informal collaboration in the sense of traditional research (Brunswicker et al., 2015; Matei, 2014; nanoHUB.org, 2014). Therefore, two developers or authors i and j will be connected if they work on a particular tool or paper in the nanoHUB.org cyberinfrastructure. However, the magnitude of connection will depend on the level of work (or intensity of digital practice activities) they put in the tools or papers. To calculate the level of interaction (digital practice capital) between any two developers or authors we apply gravity model following Matei et al. (2015) digital practice capital model. The authors applied gravity model on the basis that two scientist digital practice activities could be likened to gravitational interaction that is influenced by mass and distance as described by Isaac Newton's law of gravity

(Anderson, 2010). The authors noted that developers and authors attract with each other when working on a common tool or paper and the level of attraction is based on the amount of work (time) they put on the tools and papers. The scientists are separated by a revision distance which is defined as decayed time of association (Matei et al., 2015). The authors calculated the magnitude (weights-Θ) of the level of integration following gravity model as,

1) $$\Theta_{ij} = \frac{\delta_i \delta_j}{d_{ij}^2}$$

Where,

Θ_{ij} is the interaction term (weight) between i and j

$\delta_i \delta_j$ are functions representing attractiveness (maximum of added and deleted lines) and repulsive forces (half of the minimum added and deleted lines plus modified lines) and,

$d_{i,j}^2$ is the revision distance defined as decayed time of association.

The weights were used to construct the edge list and adjacent weight matrix of developers' collaboration in the two networks (Matei et al., 2015; Brunswicker et al., 2015).

The global network effect of the weight matrix was captured using autocorrelation modelling that is able to account for spillover effects of the network to participating scientists in addition to looking at different aspects of local network characteristics i.e., network embeddedness characteristics. Developers that are surrounded by those that have more digital practice capital will be influenced positively to be also productive because of spillover effects or contagion (Leenders, 2002). Moreover, developers that are embedded

in more than one network have more access to more digital practice capital (production resources and spillover effects) and will be expected to be more productive (Brunswicker et al., 2015; Matei, 2008). We therefore hypothesis that the network multivariate dependent and independent variables will be positively correlated to the developer's productivity. i.e.,

Hypothesis 1: Developers and Authors network aggregate digital practice capital will be positively correlated to developer's productivity.

3.2.2 Network Local Effects on Developer's Productivity

Positioning (centrality measures) and density aspects of social embeddedness are important dimension of network embeddedness that influence performance (or the level of digital practice capital) but as Gonzalez-Brambila et al. (2013) puts it, until now there is no compelling evidence of what type of network embeddedness characteristics enhance the generation of knowledge or performance. There are two main opposing school of thoughts as to what network mechanisms enhance productivity. One school of thought posits that network closure leads to more outcomes while the other posits that structural hole in network hence positioning in the network enhances better outcomes (Burt, 1992; Coleman, 1988). A third emerging school of thought argues that the "type "of scientist that one associates with might influence the outcome (Gonzalez-Brambila et al., 2013).

Gonzalez-Brambila et al. (2013) noted that network characteristics that enhance coordination include trust and this can be tested with reciprocity where members that trust each other have tendencies to reciprocate. The authors further noted that most empirical studies have focused on the structure of the network and largely ignored the effects of the relational dimension of the structure of network that include different

measurements of social capital e.g., relational capital and structural capital. The social capital deals with the importance of relationships as resources for social action (in networks) but it is not one-dimensional because different aspects of these social relationships coexist in these networks (Macke & Dilly, 2010).

Relational dimension includes centrality measures such as degree, closeness, betweeness, eigen vector centrality amongst others (Gonzalez-Brambila et al., 2013; Li et al., 2013). Degree centrality quantifies the number of direct ties that a developer has in the network (Jackson, 2008; Valente et al., 2010). It is assumed that direct ties stimulate combination and exchange of resources that are vital for accumulation of digital practice capital (Brunswicker et al., 2015; Jackson, 2008; Matei, 2014). We therefore hypothesis that developers with high degree centrality in either developer or authorship networks will consolidate resources that helps them accumulate digital practice capital to develop or author many scientific artifacts which increase their chances of getting a tool cite. i.e, *Hypothesis 2: Degree centrality will be positively correlated with developer's productivity.*

Closeness centrality measures the average distance of a developer to all others in the network (Jackson, 2008; Valente et al., 2010). A related centrality measure is betweeness centrality. Betweeness centrality measures a developer's relative position in spanning the structural hole (Jackson, 2008; Valente et al., 2010). The hypothesized effects of the two centrality measures can be closely related to density of the network whereby, for example, a denser network will lead to high measure of closeness but low betweeness centrality. The effects of density on productivity is divided nevertheless: Coleman (1998), Hansen (1999) and Obstfeld (2005), Uzzi (1997) argued that denser

networks facilitates access to information and knowledge because actors develop trust and share customs of behavior which outweigh the potential individual opportunistic behaviors amongst actors i.e., closeness centrality will give a developer a higher digital practice capital that will increase the probability of developing a tool that will get a cite. The authors concluded that denser networks are therefore more beneficial than less dense network because they increase developer digital practice capital that enables him/her to transfer tacit knowledge based on proximity. An opposing view point is that by Burt (1992, 2004), Hargadon (2002) and Hargadon and Sutton (1997) who argued that such information is likely to get redundant after sometime and that developers in less dense networks create digital practice capital through leveraging efficient and information-rich network because redundant partners is minimized. i.e., betweeness centrality will give a developer digital practice capital leverage to develop artifacts that have a higher probability of being cited. The authors found that structural holes facilitate development of innovative products. Following these constructing views we will hypothesize the two centrality measures to take any but opposite directions in the digital platform.

Hypothesis 3: Betweeness Centrality will be positively correlated to developer's productivity and Closeness Centrality will be negatively correlated to developer's productivity.

Hypothesis 4: Closeness Centrality will be positively correlated to developer's productivity and Betweeness Centrality will be negatively correlated to developer's productivity

Eigenvector centrality measures the developers relative position to more influential (powerful) or accomplished developers (Jackson, 2008; Valente et al., 2010).

It is hypothesized that a developer's association or connection with such highly accomplished (cited) developer's will enhance his/her ability to take complex ideas and thus give him/her an edge in accumulating digital practice capital that will enable him/her to develop tools or coauthor papers that have a high probability of getting a cite. We therefore hypothesize that Eigen vector centrality will be correlated to developer's productivity. i.e.,

Hypothesis 5: Eigen Vector Centrality will be positively correlated to developer's productivity.

3.3 Methodology

3.3.1 Data

The data for this study came from the nanoHUB.org cyberinfrastructure (Matei, 2014; nanoHUB.org, 2014). (Please refer to the description of the nanoHUB.org cyberinfrastructure on section 1.1.4.2 "Study Platform"). Our data from nanoHUB.org is organized by the structure of scientists that form the platform. This includes data on tool developers, tool users, educators and leaners. The data for this study comprised the tool developers and authors and this was mined from the SVN (Software Versioning Files) logs in the nanoHUB.org cyberinfrastructure (Brunswicker et al., 2015; Matei, 2014; nanoHUB.org, 2014).

3.3.2 Variables

The number of citations a developer gets from developed scientific artifact was used as the dependent variable. We do not include a time lag between the time that a developer worked on a scientific artifact and the time it was cited like in other citation studies (e.g.,

Abassi et al., 2012; Li et al., 2013) because the work on tool development involves ongoing modifications, deletions and addition of codes that are captures in the SVN logs (Matei, 2014; NanoHUB.org, 2014). The independent variables for the autocorrelation model included the weight matrices in both network spaces, network embedded characteristics that captured the local effects and control variables. The weight matrices were excluded from the network fixed effect models.

3.3.2.1 The Weight Matrices.

The gravity model weights were calculated using Equation 1 (Section 3.2.1). These weights were used to construct the edge list and adjacent weight matrix of scientist's collaboration in the two networks (Matei, 2014). The two weight matrices are presented in Figure 3 below.

Figure 3 shows developers weight matrix to be fully connected but not the authorship network. The figure also shows that developers are more than the authors. The variations could be explained by the obvious digital practice work and infrastructure involved in tools development and authorship where tool development only requires a computer that is connected to the internet to form linkages while authorship requirements and numbers are quite the opposite nevertheless (Abassi et al., 2011; Gonzalez-Brambila et al., 2013; Matei, 2014).

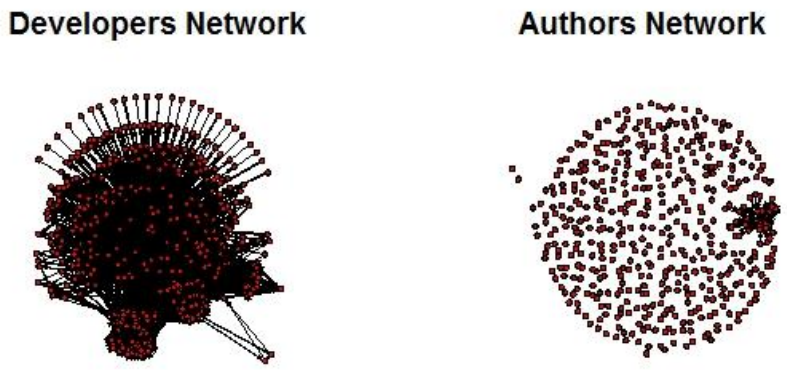


Figure 3: NanoHUB.org Developer and Authorship Networks Adjacency Weight Matrices)

The network embedded variables considered included.

3.3.2.2 Degree Centrality(C_D).

This measured the number of developers that a developer is connected to and it is calculated as,

$$C_D = d(n_i) = X_{i+} = \sum_j X_{ij} \qquad 2)$$

$d(n_i)$ is the degree centrality of node (developer) i, X_{ij} is the incoming or outgoing tie from developer i to developer j. Degree centrality is a local measure of direct contacts and its magnitude can be misleading nevertheless (Jackson, 2008.p.38; Valente et al., 2010)

3.3.2.3 Closeness Centrality($\boldsymbol{C_C(n_i)}$).

This measures how a developer is close to others in the network (Jackson, 2008.p.39; Valente et al., 2010). The measure is founded on the inverse distance of each developer to all others in the network.

3) $$C_c(n_i) = \left[\sum_{j=1}^{g} d(n_i, n_j) \right]^{-1}$$

$d(n_i, n_j)$ is the distance between developer i and j .

A developer is considered significant if he/she is relatively 'close' to all other developers i.e., has a high closeness centrality (Jackson, 2008.p.39; Valente et al., 2010)

3.3.2.4 Betweeness Centrality-$\boldsymbol{C_B(n_i)}$.

This measures the developer's ability to span structural holes (Jackson, 2008.p.39; Valente et al., 2010). The measure tallies the number of shortest paths between developers *i* and *k* that developer *j* resides on

4) $$C_B(n_i) = \sum_{j<k} g_{jk}(n_i) / g_{jk}$$

Where g_{jk} = the number of geodesics connecting *jk,* and

$g_{jk}(n_i)$ = the number that developer *i* is on (Jackson, 2008.p.39; Valente et al., 2010).

3.3.2.5 Eigen Vector and Bonacich Centrality($\boldsymbol{C(\alpha, \beta)}$).

Both centrality measures are related and they measure power (influence). Developer's "centrality (prestige) is equal to a function of the prestige of those they are connected to" (Jackson, 2008.p.40-43; Valente et al., 2010). Thus, developers that are linked to very

central developers have a higher power/prestige centrality than those who are not. The centrality measure is calculated as

5) $$C(\alpha, \beta) = \alpha(I - \beta R)^{-1} R1$$

Where, "α is a scaling vector, which is set to normalize the score, β reflects the extent to which one *weight* the centrality of developers that a developer is tied to, **R** is the adjacency matrix (can be valued), **I** is the identity matrix (1s down the diagonal) and **1** is a matrix of all ones" (Jackson, 2008.p.40-43). The author notes that the magnitude of β echoes the circle of influence/power and this distinguishes between the two centrality measures. According to Jackson, small values of β measure local structure while larger values yield global structure. i.e., If $\beta > 0$, a central developer is expected to have a high centrality when connected to other central developers and if $\beta < 0$ if the developer has a high centrality measure when connected to periphery developers. Where $\beta = 0$, the formula collapses to degree centrality (Jackson, 2008.p.40-43).

3.3.2.6 The Control Variable

Tenure. Tenure was defined as the duration of work days after a developer joined the platform

3.3.3 The Models.

This study seeks to evaluate the effect of networks in the nanoHUB.org cyberinfrastructure on developer's output. Developers in the nanoHUB.org cyberinfrastructure are embedded in developer's network but some are also embedded in authorship network. To evaluate the effect of authorship network on developers that are

embedded in the two network spaces we applied autocorrelation model, network fixed effect model and an autocorrelation model with fixed effect variable (i.e., Spatial Durbin Error Model-SDEM, probit and interaction model and fixed effect spatial probit model). Networks affects developers through structural characteristics (embeddedness) and spillover effects from the entire network (Brunswicker et al., 2015; Leenders, 2002; Matei, 2014). Embeddedness is usually considered as local network feature and global effect as spillover effects for the entire network (Jackson, 2008; Lesage & Pace, 2009). Our study will therefore evaluate both local and global effect of digital communities to participating developers. The local and global effects of network to participating developers and the effect of the number of network spaces (communities) that a scientist is embedded into in a digital platform will be modelled through network autocorrelation models (Spatial Probit and Spatial Durbin Error Models) and network fixed effect models (probit and Interaction). The fixed effect models will be used to account for the authorship network effect on developers. The models are discussed in details below: Autocorrelation models are discussed first followed by the network effect models.

3.3.3.1 Network Autocorrelation Model

The similarity in social networks and geodistance analysis is found in the weight matrix that captures the relationship in the research units while the main difference comes from the assumptions that are made regarding the research units “stationerity” (Páez, Scott, & Volz, 2008). Geodistance spatial analysis uses mostly geographical locations/features that are assumed to be stationery while the social networks use interactions mostly by humans

who are usually non-stationery because subjects change behavior quite often. The non-stationery assumption leads to measurement errors and autocorrelated error term (Dubin, 1998). However, our analysis considered a pure network effect (based on digital practice in digital ecosystem) that has less interactions of humans and it is therefore assumed to be stationery (stable) as any other geographic feature (e.g., Jackson, 2008; Orlikowski; 2000).

Our analysis involves developers that are embedded in one or two network spaces (developer /and citation network). Because we are interested in quantifying the local and global impact in two network spaces (both the developer and citation networks) we choose to extend a spatial durbin error model (SDEM) model that captures both the local and global spillovers and through the error term (Lesage & Pace, 2011). The global spillover effects are those associated with spatial lags while the local spillovers are those associated with changes in the explanatory variables (Lesage & Pace, 2011). The authors noted that one main advantage of SDEM over the conventional higher order SAR model is its ability to allow separation of the local impacts on the two network spaces (developer and authorship network) on developer' productivity. Moreover, higher order SDEM is also able to address the pitfalls associated with lack of separation of marginal effects of higher order SAR. The basic and extended SAR and extended SDEM spatial econometrics models are shown in equations 6, 7 and 8 below.

6) $$y = \rho W y + X\beta + \varepsilon$$

7) $$y = \rho_1 W_1 y + \rho_2 W_2 y + X\beta + \varepsilon$$

8) $$y = X\beta + W_1 X\theta + W_2 X\gamma + u : \ u = \rho V u + \varepsilon$$

Where,

y is a vector of dependent variable that exhibits variations across spatial observational units.

X is a vector of explanatory variables including network embeddedness characteristics

ρ, ρ_1 and ρ_2 are the scalar parameter that measure the strength of spatial dependence with the neighbors

θ and γ are scalars that measure spillovers that impact immediate neighbors (local spillovers).

β are parameters to be estimated by either the maximum likelihood, generalized moments, Bayesian, or instrumental variable methods

$W, W_1 \; and W_2$ are weight matrices representing various relationship of actors or research units.

The choice of higher order SDEM over SAR is motivated by the drawbacks that are associated with extending the simple SAR model to higher order SAR (Lesage & Pace, 2011). Lesage and Pace (2011) identified four pitfalls associated with adding weight matrices to the basic SAR. The authors noted that proponents of that extension usually do so to account for more features of non-spatial dependence and also to "stabilize" the estimates because it is believed that the estimates are highly sensitive to the weight matrix (e.g., Badinger & Egger, 2011; Case et al., 1993). Lesage and Pace (2011) noted that the ultimate goal of applying a spatial econometrics model is to explain the effects of predictor variables on the dependent variable through the own- and cross-effects which are not explained by the extended SAR model. The authors noted that extended SAR

model results in interaction and overlap of the global spillovers in the two or higher order weight matrix spaces making the own- and cross-effects non-separable. The own and cross partial derivatives from the SAR, SEM and extended SAR models are given by,

9) $$\frac{dy_i}{dx_i^r} = (I_n - \rho W)_{ii}^{-1}\beta_r$$

10) $$\frac{dy_j}{dx_i^r} = (I_n - \rho W)_{ii}^{-1}\beta_r$$

11) $$\frac{dy_i}{dx_i^r} = \beta_r$$

12) $$\frac{dy_j}{dx_i^r} = 0$$

13) $$\frac{dy_j}{dx^{r\prime}} = (I_n - \rho_1 W_1 - \rho_2 W_2)_{ii}^{-1}\beta_r$$

Equation (9) shows the direct effect of changes in the rth explanatory variable in region i to itself, while (10) shows the indirect effect of how changes in the rth explanatory variable in region i affects other regions j in the SAR model (Lesage & Pace, 2011). The direct and indirect effect can also be calculated from the resulting nXn matrix by the average of the main diagonal elements (direct) and "the average of the cumulative sum of the off-diagonal elements" (indirect effects) (Lesage & Pace, 2011). Equation (11) represents the measure of changes in the rth explanatory variable in region i to itself while (12) shows the indirect effect of how changes in the rth explanatory variable in region i affects other regions j which is zero, in the SEM model (Lesage & Pace, 2011). Equation (13) shows the partial derivative for a higher order SAR which shows the resulting nXn matrix has both W_1 and W_2 which is a combination of the two dependence that are being modelled. As such, it is impossible to separate the spillover communication

channels that are linked with each weight matrix which was the original intention of extending the SAR model to start with (Lesage & Pace, 2011). Lesage and Pace (2011) therefore identified this as the first pitfall in modelling higher order spatial models using SAR.

Lesage and Pace (2011) also examined the second belief/motivation for extending SAR model, "the sensitivity of estimated parameters to the weight matrix". The authors noted that the marginal effects in (13) could exhibit high covariations in higher order series expansions even when there was no relationship to start with. The authors noted that there might also be issues to do with endogeneity where a second non spatial weight is used for extension because it might be highly correlated with other explanatory variables.

The third drawback for extending SAR model has to do with the feasible range of the spatial dependence parameters ρ_1 and ρ_2 (Lesage & Pace, 2011). "The minimum and maximum eigenvalues of the weight matrix W determine the feasible range of the spatial dependence parameter ρ" (Lesage & Page, 2009). Lacombe and Piras (2011) and Lesage and Pace (2011) showed that the feasible region of higher order models exhibits a "non-linear relationship between feasible values of parameters ρ_1 and ρ_2". The authors indicated that most studies modelling higher order weight matrix do not specify the parameter space while others restrict the absolute values of the two parameters to less than 1.i.e.,

14) $$(|\rho_1| + |\rho_2| < 1)$$

(e.g., Badinger & Egger, 2011; Elhorst et al., 2011; Lee & Liu, 2010; Lesage & Pace, 2011). The commonly used Generalized Methods of Moment (GMM) estimation ignores the restriction of the feasible values of parameters ρ_1 and ρ_2 in (14) but Bayesian Monte Carlo Markov Chain (MCMC) estimation could be used to impose that restriction using a Metropolis-Hasting (M-H) technique (Elhorst et al., 2011).

Lesage and Pace (2011) also noted that the order with which the weight matrix is entered matters in parameter estimates and that extended higher order SAR model implicitly assumes that $W_1W_2 = W_2W_1$ which is non-flexible.

3.3.3.1.1 Addressing Draw Backs Associated with Extended SAR Model

The main motivation of applying spatial econometrics model is to capture the spillover effects associated with interdependencies in the weight matrix. As aforementioned the spillover effects are usually local or global. Models that capture the local spillovers effects include "spatially lagged explanatory variables (SLX) and spatial durbin error (SDEM) models" but these models have been largely been ignored in applied work (Lesage & Pace, 2011). Equation (15) and (16) give the model specifications,

15) $$y = X\beta + WX\theta + \varepsilon$$

16) $$y = X\beta + WX\theta + u: \ u = Vu + \varepsilon$$

All the variables are as explained in above and θ is the parameter that captures the local effects. The partial effects of (15) and (16) is the same and it given by (17),

17) $$\frac{dy_i}{dx_i^r} = (I_n\beta_r + W\theta_r)$$

The average of the diagonal in (17) gives the direct effects while the average of the off-diagonal elements gives the indirect effect (Lesage & Pace, 2011). The diagonals

elements in the weight matrix W are zeros (reflecting the fact that a region cannot be neighbor to itself) and the row sums are 1 implying that from (17) β_r gives the direct effects while θ_r gives the spillover effects of the immediate neighbors (local effects) (Lesage & Pace, 2011). The authors noted that SDEM model has also the ability to give the global effects through the error term and it is therefore more efficient. An extended/higher order SDEM model is given by,

18) $$y = X\beta + W_1X\theta + W_2X\gamma + u : \ u = \rho Vu + \varepsilon$$

Equation (18) gives separate local and global spillover effects and is able to avoid the aforementioned pitfalls of extending SAR model (Lesage & Pace, 2011). SDEM model (18) was therefore chosen for the analysis of this study. Lesage and Pace (2011) further noted that extended SAR model has the same functional form the expected y and the error term covariance which is restrictive because misspecification in one part will taint other parts of the model specification. A Bayesian Monte Carlo Markov Chain (MCMC) estimation method was applied over the commonly used Generalized Methods of Moment (GMM) to get the estimates of the SDEM model (Eqt. 18). GMM estimation ignores the restriction of the feasible values of parameters $\rho 1$ and $\rho 2$ in (14) but Bayesian Monte Carlo Markov Chain (MCMC) estimation was used to impose that restriction using a Metropolis-Hasting (M-H) technique (Elhorst et al., 2011). Bayesian estimation method samples posterior distribution parameters from our model and then applies Markov Chain Monte Carlo (MCMC), Gibbs and Metropolis-Hasting technique to generate population using several simulations (here 1000) and confidence interval with a burn.in value (here 500).

3.3.3.2 Network Fixed Effect and Interaction Models

Fixed effects regression models holds constant (fixes) the average effects of each developer and is able to capture the effect of within variation in the authorship network (Wooldridge, 2003. p.220). The modelling involves inclusion of authorship dummy that controls for the average differences across developers i.e., the fixed effect coefficient controls the variations across the developer networks and only leaves the variations within authorship network. The fixed effect probit model was used as the non-spatial version of the fixed effect spatial probit models. The fixed probit model does not include the spatial autocorrelation variable and was used to compare/ or validate the use of the spatial version. The interaction model extends the single fixed effect probit model. In the model, we assume that the authorship dummy moderates the effects of other variables too. We therefore interact the authorship dummy with the network structural variables and control variables. Interaction of the authorship dummy with continuous variables will alter the slope while interaction with dummy variable will alter the intercept (Wooldridge, 2003. p.233; Green, 2003.p.123). The interaction model was used to evaluate the effect of the network structure and control variables on scientist citations conditional (when moderated) by the developer also being an author. The fixed effects probit model and interaction models are presented in equations (19) and (20) below.

19) $$y = X\beta + \varepsilon$$

And

20) $$y = X\beta + X_{-1}X_1\theta + \varepsilon$$

Where,

y is a vector of dependent variable with 1 and 0's.

X is a vector of explanatory variables including network embeddedness characteristics

β is a vector of parameters to be estimated

$X_{-1}X_1$ is a vector of interaction variables defined as; X_{-1} is a vector of the all explanatory variables excluding the dummy of a developer being an author and X_1 is the dummy representing a developer who is also an author

θ is a vector of fixed effect parameters to be estimated

3.4 Results and Discussion

We first conducted statistical data analysis[9] visually, then tested variables for spatial effects before formal modeling and hypothesis testing. All variables were first explored visually through histograms before being analyzed statistically. Histograms of citations, tenure, centrality measures; bonacich, betweeness, closeness, degree and eigen vector and components are presented in the Appendix. Histograms for citations, betweeness centrality, degree centrality and eigen vector centrality are positively skewed and show distribution that follows power law. Table 2 show the descriptive statistics of the variables used in the model.

[9] "Statistical data analysis involves both statistical analysis and visual inspection of the variables" (Dasu & Johnson, 2003).

Table 2: Descriptive Statistics

Variable	Obs	Mean	Std. Dev.	Min	Mid	Max
Citations	477	13.94	68.99	0	0	866
Tenure	477	1878	1039	0	7.50	4974
Authorship (dummy)	477	0.612	0.488	0	0	1
Bonacich Centrality	477	-0.13	-0.99	-7.679	0.008	3.111
Betweeness Centrality	477	767.04	5077.87	0	0	67292
Closeness Centrality	477	5.15e-5	7.69e-7	4.4e-6	5.75e-5	6.14e-5
Degree Centrality	477	53.77	67.736	1	13	597
Eigen Vector Centrality	477	0.101	0.224	2.7e-4	0.02	1
Components	477	239.00	137.84	1	239	477

Table 2 shows, mean standard deviation, minimum, median and maximum of the variables. All centrality variables have high standard deviation, a median that is close to the minimum and high range (minimum and maximum difference) which of an indication of positive skewedness. The average number of citations that an article gets is about 14 with a standard deviation of 69. However, the minimum and median citations that a developer gets is zero implying that there are many developers that get very low citations and very few that get high citations. The average number of citation an author gets is within range found by Gonzalez-Bambrila et al. (2013) and Singh (2007) even though their articles were in different study areas. The average number of days of tenure that the developers had since joining the nanoHUB.org cyberinfrastructure was 1878 with a standard deviation of 1039. The authorship dummy had value 1 if the developer was also an author and 0 if the developer was not an author. Results show that about 39% of the developers were also authors. This implies that over 60% of software developers do not attempt to publish their work. The mean values of bonacich and eigenvector centrality

measures were -0.127 and 0.101 respectively. Their standard deviations were -0.993 and 0.244. These results imply that the number of influential/powerful/very successful developers' in the network is relatively small. Betweeness centrality measure also showed high variance with a mean of 767.04 and a standard deviation of 5077.87. Betweeness centrality measures show the average span across the network structural holes and the high number is an indication of that most developers have relative ease in spanning across the structural holes in the network. i.e., developer network in the nanoHub.org cyberinfrastructure has many components and good enabling mechanisms that allow developers to easily span through those components. Burt (1998) and Gonzalez-Bambrila et al. (2013) also noted that high betweeness centrality can also be attributed to the size of the network where large sample size increases the structural holes.

The mean in degree centrality measures was 53.8 with a standard deviation of 67.7. However, the minimum and median in degree was 1 and 13 respectively implying that most developers have a low number of indegree and few have a high indegree. The results are characteristics of citations network that lean towards being scale free (e.g. Barbasi & Albert, 1999; Jackson, 2008). The number of component showed relatively normal distribution with a mean of 239 and a standard deviation of 137.84. Component measures the number of developers that are "reachable from a given developer", or the opposite: all developers from which a given developer is "reachable via a directed path" (Gabor, 2014).

The variables that showed had high variance were tested for power law distribution (scale free property) through Kolmogorov–Smirnov test (KS[10] test). The power law distribution test had the null hypothesis that the data was generated from a distribution that was scale free (power law distribution). The KS test results of the power law distribution tests are shown in Table 3.

Table 3: KS Test for Power Law Distribution for Selected Variables

Variable	**alpha**	**KS.Stat**	**KS.P**
Citations	2.726	0.121	0.143***
Bonacich Centrality	3.256	0.000	1.000***
Betweeness Centrality	1.717	0.089	0.927***
Closeness Centrality	13.206	0.242	0.000
Degree Centrality	3.477	0.147	0.727***
Eigen Vector Centrality	1.736	0.094	0.003

*** denote significance at 1% significance level

Results show that the column KS p value for citations, bonacich centrality, betweeness centrality, degree and centrality were greater than 0.05 and we reject the null hypothesis that their distributions did not come from a power distribution. We therefore conclude that the data set came from power law distributions. Similar results have been found in other citation and social networks (e.g. Barbasi & Alberta, 1999; Jackson, 2008). Log transformation was applied to the power law distributed variables to correct (have more variance) for high positive skewedness in those transformations which is more suitable for parametric regression (Hoskins, 2013). The distributions of the transformed variable

[10] "Kolmogorov–Smirnov test is a nonparametric test of the equality of continuous, one-dimensional probability distributions that is used to compare a sample with a reference probability distribution for one-sample KS test." (Marsglia et al., 2003)

were further examined visually. All the variable expect degree centrality still exhibited positive skewedness. These variable were therefore categorized into two as follows; the dependent variable citations were set to 1 if the number of citations were greater than 1 and 0 otherwise. The implication of categorizing the endogenous variable was that we now have a dichotomous variable that can longer fit a linear model (Wooldridge, 2003). Running a liner model on limited dependent variable results in inefficient estimates (Lesage, 2000). For the predictor variables, we created a dummy variable for betweeness, closeness and eigen vector centrality. Betweeness centrality was set to 1 if the measure was greater than 2 and 0 otherwise. Closeness centrality measures had very low values and was therefore scaled up by 10^5 before being categorized into two; 1 was assigned if the closeness centrality was greater than 4 and 0 otherwise. The eigen vector centrality was set to 1 if the value was greater than 0.1 and 0 otherwise. The corresponding global descriptive properties of the networks are presented in Table 5. The global statistics that were considered include Assortativity, Clustering coefficient, diameter, density and reciprocity. The definition, magnitude and implication of the global statistics is discussed below.

Table 4: The Global Descriptive Properties of the Developer and Authorship Network

Variable	**Developers Network (W1)**	**Authorship Network (W2)**
Assortivity	-0.0075	-0.0026
Clustering coefficient	0.7595	0.7595
Diameter/avshortpath	2.22284	2.22284
Density	0.0565	0.0565
Reciprocity	0.3780	0.378

Both networks have equal magnitudes of clustering, density and reciprocity. Results show assortativity coefficients of both networks are low (compare -0.0075 to -0.0026 for developer network and authorship network respectively). The low coefficient implies that there is low homophily in the network because assortativity coefficient measures the tendency of scientists to mix with similar scientists in a network (homophily) (Newman, 2003). Clustering coefficient is also known as transitivity coefficient and it measures the probability that adjacent developers of a scientist are connected (Gabor, 2014; Wasserman & Faust, 1994). The clustering coefficient was 0.7595 for both networks. This implies that developers will cluster into small group than into bigger one. The networks have a low density of 0.0565 which indicates that there is a low probability of getting a tie (dyad) in a purely random network. Burt (1992, 2004) and Hagdom and Sutton (1997) found that low dense networks accord scientist leverage in generating opportunities that are more efficient and non-redundant. Both networks have reciprocity of 0.378. Reciprocity describes the proportion of mutual connections in a directed graph. i.e., "the probability that the opposite counterpart of a directed edge is also included in the graph" (Gabor, 2013). Reciprocity is often used as a measure of trust in social exchange theory (Cropanzano & Mitchell, 2005). Result imply that about there is about 37.8% probability of mutual connections or social exchange between developers in the network.

3.4.1 Statistical Test for Spatial Autocorrelation

Moran I was used to test for spatial effects; spatial autocorrelation and heterogeneity of the variables and in the model. Moran's I scatter plots were also plotted to visually

explore the autocorrelation patterns in the variables. Results for Moran's I results are presented in Table 5 while some representative scatter plots are presented in the Appendix.

3.4.1.1 Moran's I Test for Spatial Autocorrelation

Moran's I test for residuals is given by,

21) $$I = [N / S]\{[\varepsilon' W \varepsilon'] / \varepsilon' \varepsilon\}$$

Where ,

ε is the vector of residuals

W is a exogenous spatial weight matrix defined above and

S is a standard factor defined as the sum of all elements in the given matrix (Anselin, 1988).

Moran's I test for residuals had the null hypothesis of no spatial effects on the endogenous and predictor variables. Moran's I test for the residual were tested under the assumptions that the distribution of the variables was normal and random pattern but both yielded similar results (Tiefelsdorf, 2000). Moran's I results are presented in Table 5 below,

Table 5: Moran's I Statistics for Dependent and Independent Variables

Variable	**Developer Network (W1)**		**Authorship Network (W2)**	
	Moran I	**p-value**	**Moran I**	**p-value**
Citations	0.223***	0.000	0.940***	0.000
Bonacich Centrality	0.125***	0.000	0.940***	0.000
Betweeness Centrality	0.256***	0.000	0.942***	0.000
Tenure	0.137***	0.000	0.939***	0.000
Closeness Centrality	0.149***	0.000	0.938***	0.000
Degree Centrality	0.177***	0.000	0.942***	0.000
Eigen Vector Centrality	0.171***	0.000	0.939***	0.000
Google Page Rank	0.178***	0.000	0.943***	0.000

*** denote significance at 1% significance level

Results show that we reject the null hypothesis of no spatial effects in the dependent and independent variables in both weight matrices. Results also show that the slopes of the fitted line in the second weight matrix is higher than the first weight matrix. The scatter plots in the Appendix seem to support Moran's I test statistics results: the plots show clear patterns of clustering along the fitted line in the quadrants for all the variables and the slope of fitted line in the authorship network is higher.

Table 6 show the correlation matrix between the variables used in the models. The variables are presented symbols as follows: Citations (y), Tenure (x02), Developers network variables (W1-Weight matrix 1); Bonacich centrality (x03), Betweeness Centrality (x04), Closeness Centrality (x05), Degree Centrality (x07), Contributions (x08), Eigen Vector Centrality (x09), Components (x10) and Authorship network variables (W2-Weight Matrix 2); Bonacich centrality (xx03), Betweeness Centrality (xx04), Closeness Centrality (xx05), Degree Centrality (xx07), Contributions (xx08), Eigen Vector Centrality (xx09), Components (xx10).

Table 6 shows that some variables have high correlation. The table shows Bonacich and betweeness centrality from the two weight matrix as being perfectly correlated. Betweeness centrality in the two weight matrices has also a high correlation. The highly correlated variables were removed from the model before analysis[11].

Table 6: Correlation Matrix of Dependent, Control and Network Structural Variables Considered in the Models

Variable	y	x02	x03	x04	x05	x07	x08	x09	x10	xx03	xx04	xx05	xx07	xx08	xx09	xx10
y	1															
x02	-0.1	1														
x03	0.0	0.2	1													
x04	0.3	0.0	0.0	1												
x05	0.0	0.1	-0.1	0.1	1											
x07	0.2	-0.1	-0.3	0.5	0.3	1										
x08	0.3	0.0	0.1	0.1	-0.1	-0.1	1									
x09	0.2	0.0	0.0	0.1	0.2	0.5	0.4	1								
x10	0.0	0.0	0.2	-0.2	-0.3	-0.4	0.2	-0.1	1							
xx03	0.0	0.2	1.0	0.0	-0.1	-0.3	0.1	0.0	0.2	1						
xx04	0.3	0.1	0.1	0.8	0.1	0.5	0.1	0.2	-0.1	0.1	1					
xx05	0.0	0.1	-0.1	0.1	1.0	0.2	-0.1	0.1	-0.3	-0.1	0.0	1				
xx07	0.2	-0.1	-0.3	0.5	0.3	1.0	-0.1	0.5	-0.4	-0.3	0.5	0.2	1			
xx08	0.3	0.0	0.1	0.1	-0.1	-0.1	1.0	0.4	0.2	0.1	0.1	-0.1	-0.1	1		
xx09	0.2	0.1	0.0	0.1	0.1	0.3	0.3	0.6	0.0	0.0	0.2	0.1	0.3	0.3	1	
xx10	0.0	0.1	0.3	-0.2	-0.4	-0.7	0.2	-0.3	0.4	0.3	-0.2	-0.3	-0.7	0.2	-0.1	1

3.4.2 Models Results and Discussion.

The network autocorrelation model and the fixed effects network models are presented in Table 7 below. The models fit test statistics are also presented in the Table 7. Table 7 show results of fixed effect network (probit and interaction) models and network

[11] We use R programming software which does not run if there is collinierity problem. R program does not invert the matrix X^TX that is used to generate the estimate. A consequence of running the model with collinear variables is getting high standard error that increase the probability of causing type two error and getting over fitted models

autocorrelation (spatial probit and spatial durbin error-SDEM) models. The models were first subjected to likelihood ratio (LR) tests to evaluate their fit of the data. The LR test had the null hypothesis that the log likelihoods of restricted and unrestricted models are not different from zero. Results show that log likelihood of the restricted and unrestricted models are all different from zero and therefore all the four models fit the data presentation. Given the LR test results that qualify all models, we use a vote count technique that is often used in meta-analysis[12] to discuss the models estimates results. Vote count method is a simple narrative review in which the number of statistically significant studies are compared to the number of statistically non-significant studies using p-values (Stanley & Doucouliagos, 2012). We therefore proceed to discuss the direction and magnitude of variables based on the vote count in the four models.

Table 7 shows the spatial autocorrelation parameter of the spatial probit model is statistically different from zero. The parameter results of -0.003 at 1% significance level implies that there is a negative spatial spillover effect in the developer network. Being embedded in the developer network reduces the probability of developing a tool that will get a cite.

[12] "Meta-analysis synthesizes results from a group of studies while controlling for heterogeneity among studies and consequently builds a body of knowledge and that provides a more precise and robust guide for action" (Stanley & Doucouliagos, 2012, p. 3; Ringquist, 2013, p.4).

Table 7: Regression Results for Network Effect (Probit and Interaction) Models and Network Autocorrelation (Spatial Probit and SDEM) Models: DV=Number of Citations of Scientific Artifacts

	PROBIT	**SARPROBIT**	**INTERACTION MODEL**	**SDEM (Extended)**
Dependent Variable (DV)=Y	**Estimate (std dev)**	**Estimate (std dev)**	**Estimate (std dev)**	
Intercept	0.122 (0.075)	-1.823** (0.698)	0.020 (0.086)	-0.025 (0.715)
Tenure	-0.008 (0.009)	-0.059 (0.084)	-0.003 (0.009)	0.133 (0.079)
Betweeness Centrality (Dummy)	0.016 (0.029)	0.136 (0.232)	-0.000 (0.038)	0.0451** (0.170)
Closeness Centrality (Dummy)	-0.115** (0.042)	-1.145** (0.350)	-0.001 (0.054)	-0.521** (0.217)
Authorship member (Dummy)	0.775*** (0.025)	3.627*** (0.279)	0.925*** (0.076)	
Degree Centrality	0.007 (0.010)	0.083 (0.104)	-0.000 (0.011)	-0.161* (0.093)
Eigen Vector Centrality	0.198** (0.039)	1.142** (0.383)	-0.000 (0.048)	-0.439* (0.234)
Components				-0.000 (000)
Authorship member (Dummy) and Betweeness Centrality			0.011 (0.059)	
Authorship member (Dummy) and Closeness Centrality			-0.253** (0.086)	
Authorship member (Dummy) and Degree Centrality			0.012 (0.023)	
Authorship member (Dummy) and Eigen Vector Centrality			0.252** (0.082)	
Bonachis Centrality (local_1)				0.000 (0.004)
Degree Centrality (local_1)				-0.116 (0.219)
Tenure (local_2)				-0.154** (0.065)
Bonacich Centrality (local_2)				-0.030 (0.067)
Betweeness Centrality (local_2)				0.871*** (0.176)
Components (local_2)				0.001 (0.001)
AIC	50.3	220.318	34.213	550.21
rho		-0.003*** (0.001)		-0.305 (0.387)
Sige				1.062*** (0.113)
Log-likelihood		102.16	-5.11	-260.11
LR (nested interaction terms)	543.76***	352.84***	24.09***	31.06***

*, ** and *** denote significance at 10%, 5% and 1% significance level, respectively

Both weight matrices are characterized by high clustering[13] (small worlds) but do not show homophily amongst those clusters (low assortativity coefficient). This implies that developers in both developer and authorship networks cluster not based on similarity in them (scientists) but other factors that could be work related[14]. The weight matrices are also characterized by low density and relatively low reciprocity. High clustering, low density and reciprocity will encourage developers to span structural hole while searching for non-redundant knowledge from "trusted" (reliable) scientist that will give them leverage to develop quality tools that have a high probability of getting a cite (e.g., Burt 1992; 2004; Hargadon & Sutton, 1997). These results are further supported by the positive and significant coefficient of betweeness centrality and negative and significant coefficient of closeness centrality. These findings support Burt (1992; 2004) and Hargadon and Sutton (1997) structural hole theory as the mechanism that enhances developer's productivity in a digital network. The results also support our hypothesis of getting reversed influence between closeness and betweeness centrality measure in digital network.

The structural network characteristics of eigen vector centrality and closeness centrality have statistically significant effects on probability of developing a quality tool that will get citations in the probit, spatial probit and SDEM models. Eigen vector centrality measures the developer's position relative to influential/highly accomplished developers in the network. Results show that being close to influential developers in the

[13] Clustering coefficient for both weight matrices was 0.76 while assortativity (homophily) measure for weight matrices 1 and 2 was calculated as -0.0075 and -0.0026 respectively (See Table 4 in section 4.2)

[14] Our data lacks more control variables that could explain the behavior better.

network increases the probability of getting a citation. This result is supported by Gonzalez-Bambrila et al. (2013) but contradicted by Abbasi et al. (2011). Abbasi et al. (2011) found a negative correlation between eigen vector centrality and scientist's productivity. However, the findings support the emerging new school of thought which argues that the "type" of scientist that a developer associates/works with might influence citation of developed tools (Gonzalez-Bambrila et al., 2013). Being close to other developers in the network leads to reduced probability of developing a tool that will get a cite. These results are supported by Coleman (1988), Burt's (1992&2004) and Hargadon and Sutton (1997) who argue that high closeness centrality leads to redundancy in information or knowledge. This result is confirmed by statistically significant negative estimates of closeness centrality measure in all the models but interaction term model. Tools specificity that a developer works on could be attributed to specialization and redundancy in ideas.

The dummy variables for a developer being an author (authorship dummy) yield statistically significant results for both the probit and spatial probit models. Authorship dummy was used to evaluate the effect of a developer also being an author after fixing the effect of developers in all models but SDEM model. Results show that the authorship dummy was positive and statistically significant at 1% significance levels. Results show that a developer who is also an author has about 77.5%, 92.5% and over 100% likelihood of developing a tool that will get a cite going by the probit, interaction and spatial probit models respectively. These results imply that being embedded in multiple networks increases the chances of developing a tool that will get citation.

The interaction terms in the interaction model show that closeness centrality reduced the probability of a developer who is an author from getting citation of their tool after moderating the effect of embeddedness in the author network. i.e., high closeness centrality reduces the probability of a developer developing a tool that will get a cite when the scientists is a developer and an author. However, eigen vector centrality increases the probability of getting a citation when it is moderated in the authorship network. The SDEM model results show that scientists that span structural holes (have high betweeness centrality) have a higher probability of developing a tool that will get a citation. This result is supported by Burt (1992; 2004) that showed that structural hole facilitates development of quality tools that will most likely get citations. SDEM model result also show that closeness, degree and eigen vector centralities have a negative influence in developing a tool that will get cited. Developers with low degree centrality have a higher probability of developing a tool that will get a cite. This finding goes against our hypothesis but it can be attributed to the above highlighted tool specificity and the tendency for information to became highly redundant amongst many scientist working on one particular too (e.g. Burt 1992; 2004). The tenure and betweeness centrality local spillover effect in the authorship network have significance influence of development of a tool that will get citation. Longer tenure reduces the probability of getting a tool that will get citation while betweeness centrality increases the probability of getting a tool that will get citation.

3.5 Summary and Concluding Remarks

This study sought to evaluate network structural and relational factors that influence developer's productivity in the nanoHUB.org cyberinfrastructure. Our study evaluated developer's productivity and we hypothesized that productivity in a digital ecosystem is a function of the network (structural and relational features) and developer's inherent characteristics including the number of networks that a developer is embedded into. Data for this study came from the nanoHUB.org cyberinfrastructure. The nanoHUB.org has data that is organized by the structures of scientists in the platform and includes data of tool developers, authors, tool users, educator and learners.

A network and an extended (Spatial Probit and Spatial Durbin Error) network autocorrelation models were used to capture both the network relational aspects (spillover effects) and also embeddedness in multiple network spaces in the digital platform. The model results were compared to network fixed effect models (probit and interaction models). The number of citations a developer gets from the developed tool was used as the dependent variable. The independent variables for the autocorrelation model included the weight matrices in both network spaces, network embedded characteristics that captured the local effects and control variables.

Results showed that the spatial autocorrelation parameter of the spatial probit model is statistically different from zero. The parameter results of -0.003 at 1% significance level implies that there is a negative spatial spillover effect in the developer network. i.e., being embedded in the developer network reduces the probability of getting a citation. Results of the extended SDEM also show a negative but statistically insignificant spatial effect parameter. The results contribute to both theoretical and

practical understanding of networks where autocorrelative modelling is extended to understand the effects of networks formed in digital practice. The negative spillover effect was attributed to model representation and the characteristics of the chosen weight matrix/matrices. Both weight matrices are characterized by high clustering[15] (small worlds) but do not show homophily amongst those clusters (low assortativity coefficient). The practical implication of these results is the revelation that developers in both developer and authorship network cluster not based on similar developers but other factors that could be work related. The weight matrices were also characterized by low density and relatively low reciprocity. High clustering, low density and reciprocity encourages developers to span structural hole while searching for non-redundant knowledge from "trusted" (reliable) developers that will give them leverage to develop quality tools that have a high probability of getting a cite (e.g., Burt 1992; 2004; Hargadon & Sutton, 1997).

The structural network characteristics of eigen vector centrality had statistically significant effects on probability of getting citations. Eigen vector centrality measures the developer's position relative to influential/highly accomplished developers in the network. Results showed that being close to influential developers in the network increases the probability of getting a citation. This finding is a major theoretical contribution that supports the emerging new school of thought which argues that the

[15] Clustering coefficient for both weight matrices was 0.76 while assortativity (homophily) measure for weight matrices 1 and 2 was calculated as -0.0075 and -0.0026 respectively (See Table 4 in section 4.2)

"type" of developers that a developer associates/works with might influence development and citation of tools they develop (Gonzalez-Bambrila et al., 2013).

The dummy variables for a developer being an author (authorship dummy) yielded statistically significant results for both the probit and spatial probit models. Authorship dummy was used to evaluate the effect of a developer also being an author after fixing the effect of developers. Results show that the authorship dummy was positive and statistically significant at 1% significance levels. These results are also a major practical contribution in digital practice organization since they reveal that being embedded in multiple networks increases the chances of developing a tool that will get citation.

CHAPTER 4. GROWING DEVELOPER COMMUNITY IN SCIENTIFIC DIGITAL COMMUNITIES: EXPONENTIAL RANDOM GRAPH MODELS

4.1 Introduction

Communities in digital platforms (ecosystems) have been growing rapidly in the last two decades mainly due to improvement in computing technologies (Schroeder et al. 2007). The growth of these communities has made the platforms an important part of the collaboration infrastructure of the current society. The growth has also seen an equal increase in studies researching the patterns of formation and sustenance of these communities (e.g., Crowston et al., 2012; Rossi, 2006; Scacchi, 2007). The majority of this literature is comprised of studies that have modelled online communities as networks that are formed by actors who form and break ties (collaborate) in those environments based on their inherent goals (e.g., Jarvenpaa & Leidner, 1999; Kanawattanachai & Yoo, 2007; Kankanhalli et al. 2005; Wasko & Faraj, 2000&2005). Others have looked at the effect of the networks on community members' outcome (e.g., Abbasi et al., 2012; Brass, 2002; Brunswicker et al., 2015; Gonzalez-Brambila et al., 2013; Matei, 2014). However, most of these studies only describe and understand the network characteristics and their effects on community member's outcome; they rarely address the mechanism of network formation (Jackson & Rogers, 2007; Robins et al., 2007). Research in digital platforms consists of social networks that are enabled by computer systems that are linked by internet. The social networks provide an informal system of social and technical

interactions which facilitate scholarly scientific collaboration from the digital practice related activities such as software development (Brunswicker et al., 2015; Kling et al., 2003; Matei, 2014). This new form of technology enabled networks has been growing at unprecedented rate but there is limited knowledge of how the networks form and how they sustain themselves (Faraj & Johnson, 2011; Jackson & Rogers, 2007). For example, "nanoHUB.org cyberinfrastructure user community grew from 1,000 in 2002 to more than 56,000 in 2007 while 5,800 registered users logged in and ran more than 240,000 simulation jobs in 2007" (Klimeck et al., 2008). This study will therefore seek to understand the network formation and sustenance mechanism in an online community (nanoHUB.org cyberinfrastructure) through social network modelling. Robins et al (2007) gave several reasons as to why we would need to model network formation over and above the well-known and applied techniques that measure properties of network on outcome; The authors noted that modelling networks gives a better understanding of the social behaviors responsible for predominantly self-organized network formation processes given that the social behavior is complex as it involves aspects of both randomness and regularity. Robin et al. (2007) also noted that statistical modelling yielded better inference about which aspects of network are more prevalent than they would be expected in say a purely random or preferential formation process. The authors further noted that modelling allows us to better understand which social network processes might be dominant in explaining a phenomenon like clustering that is usually caused by either endogenous structural effects (self-organization) or node level effect (e.g. homophily).

This study therefore models social network in the nanoHUB.org cyberinfrastructure so as to understand the mechanisms of organization in this emerging organization structures of developer's communities. The study is broadly guided by the research question; What are the network formation and sustenance mechanism and structural characteristics of a digital platform? This study contributes to the emerging literature of understanding the virtual organizational of large communities of developers. To the network formation and organization mechanisms, study draws upon theories of network, network exchange theory, preferential and random networks formation theory and mutual interest/collective action theory (Albert & Barbasi, 1999; Borgatti & Halgin, 2011; Faraj & Johnson, 2011; Jackson & Rogers, 2007; Monge & Contractor, 2003).

4.2 Theoretical Framework and Hypothesis

4.2.1 A Framework of Network Formation through Digital Practice in Community of Developers

Network formation process falls broadly under the theory of network, where individual's inherent characteristics/attributes are assumed to influence the type of ties they form and with whom (Borgatti & Halgin, 2011; Jackson, 2008). Theory of network is defined as the study that determines why network form. i.e., models of which actors form ties (links, triads, e.t.c) and how do they position themselves (e.g., centrality measures, small-worldness e.t.c) the network as a whole will have (Borgatti & Halgin, 2011). Matei (2014) explained the individual motivation and inherent characteristics to joining and participating in these platforms can be explained and revealed by the digital practice theory that argues that evolution of networks is caused by the level and intensity of

digital activities that the members engage in. In addition to the above two theories: theory of network and theory of digital practice we will also consider several complementing theories (sub-theories) have also been used to explain the growth and sustenance of networks in online communities. These theories include network exchange theory, preferential and random networks formation theory and mutual interest/collective action theory and could be looked at as those that explain the mechanism that is holding the network in place (Albert & Barbasi, 1999; Faraj & Johnson, 2011; Jackson & Rogers, 2007; Monge & Contractor, 2003).

While modelling network formation from a network level characteristic this study will be able to reconcile conflicting motivations for why developers join the communities and also understand the network-level characteristics that are responsible for growth and sustenance of online communities like the nanoHUB.org cyberinfrastructure. Research on collaboration in online communities has also identified developer's inherent characteristics that are motivated by self-gain for reputation building and pure altruism as some of the reasons that developer's participate in the platforms (Constant et al., 1996; Fulk et al., 2004; Matei, 2014; Peddibhotla & Subramani, 2007). In online communities, software developers collaborate virtually through digital practice activities in the digital platform (Matei, 2014; nanoHUB.org, 2014). The digital practice work on the tools involve modification, deletion or addition of the original software code and this is captured in the subversion (SVN) logs (Matei, 2014; nanoHUB.org, 2014). SVN is version control that assists manage changes to the tools source code and therefore enables us to capture the intensity of digital practice activities (Cambridge, 2015). SVN control/registry manages the changes by preventing programmer developers that are

working on the same source code, from "overwriting" each other's codes, "possibly reintroducing bugs some poor programmer has spent ages removing." SVN works like central repository, but it "remembers every change ever made to the files and directories" (Cambridge, 2015). This allows recovery and examination of the history of changes (including how and when the data was changed and who changed it) of older versions of a developers file (source code lines) "you to recover older versions of your files and examine the history of how and when your data changed, and who changed it". The nanoHUB.org allows multiple and parallel modification of source code in a copy-modify-merge SVN management system. The copy-modify-merge SVN management system in illustrated and explained below.

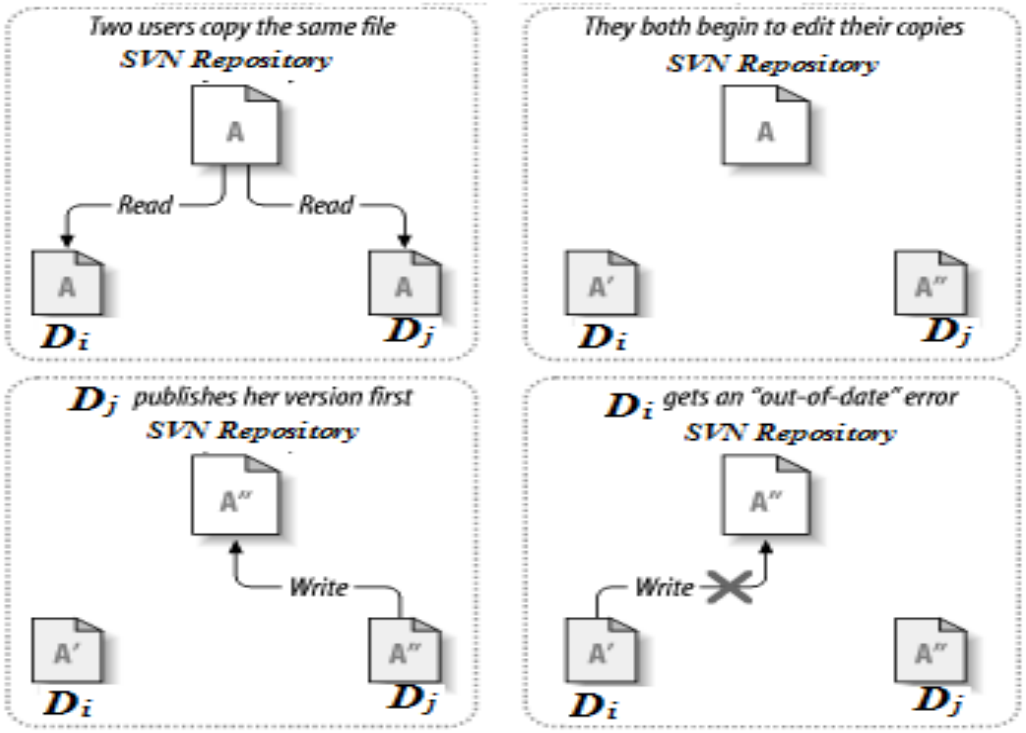


Figure 4: Illustration of Copy-Modify-Merge Subversion Management System

The sequence of changing a source code (e.g., a code line, function) in the copy-modify-merge version in Figure 1 involves the following processes (Cambridge, 2015; nanoHUB.org),

- Developers *i* and *j* "each create working copies of the same source code, copied from the SVN repository".
- Both developers work in parallel, and modify the same code (e.g., source code line "A".
- Developers *j* saves her modifications to the repository first.
- Developers *i* attempts to save his modification thereafter, but the repository "informs him that his source code file A is out-of-date; file A in the repository has somehow changed since he last copied it."
- So Developers *i* asks "his client to merge any new changes from the repository into *his* working copy of file A (it is assumed here that there are no conflicts)".
- Both sets of modifications are integrated, and Developers *i* "saves his working copy back to the repository."

This makes developers *i* and *j* connected by the virtue of working on a similar source code in the platform. However, the magnitude of connection will depend on the level of work they put in the tools or papers. To calculate the level of interaction (digital practice) between any two developers we apply gravity model following Matei et al. (2015) digital practice proximity model. The authors applied gravity model on the basis that two developers digital practice activities could be likened to gravitational interaction that is influenced by mass and distance as described by Isaac Newton's law of gravity

(Anderson, 2010). The authors noted that developers attract with each other when working on a common tool and the level of attraction is based on the amount of work (time) they put on the tools. The scientists are separated by a revision distance which is defined as decayed time of association (Matei et al., 2015). The authors calculated the magnitude (weights-Θ) of the level of integration following gravity model as,

22) $$\Theta_{ij} = \frac{\delta_i \delta_j}{d_{ij}^2}$$

Where,

Θ_{ij} is the interaction term (weight) between i and j

$\delta_i \delta_j$ are functions representing attractiveness (maximum of added and deleted lines) and repulsive forces (half of the minimum added and deleted lines plus modified lines) and,

$d_{i,j}^2$ is the revision distance defined as decayed time of association.

The weights were used to construct the edge list and adjacent weight matrix of developer's collaboration in the developer networks (Matei et al., 2015). The growth and attachment patterns in online communities has been studied from the network formation perspective, where developers are believed to have some preferences while attaching (contributing to source code) to other actors in the network (e.g., Barbasi & Alfred, 1999). Research of theory of network has mostly involved evaluation of the network formation processes as either random (e.g., Erdos Renyi, ERGM-Exponential Random Graph Models), preferential (preferential attachment models that have distributions that are scale free –Barbasi & Alfred, 1999) or hybrid model involving both processes (Jackson, 2008). These models evaluate the network from the actor's behavioral point of

view i.e., by looking at models of which actors form ties (links, triads, e.t.c) and how do they position themselves (e.g., centrality measures, small-worldness e.t.c) the network as a whole will have due to their action (Brass, 2002; Jackson, 2008). Recent network formation studies have found that actors do not follow preferential attachment while joining a group but do so randomly (e.g. Jackson & Roger, 2007; Faraj & Johnson, 2011). Based on those contrasting viewpoints, this study hypothesis that developers in digital platforms exhibit both preferential and randomness searches while looking on what source code they want to contribute to.

HYPOTHESIS 1a: Developers in digital platforms contribute to source code randomly.

HYPOTHESIS 1b: Developers in digital platforms contribute to source code preferentially

4.2.2 A Framework of Network Efficiency and Sustenance: Network and Social Exchange Theories

Network exchange theory posits that people have different levels of resources and can exchange them based on their desire which is also the case with developers that have different levels of expertise (Monge & Contractor, 2003). The theory also states that the structure of the network constraint drives different developers to act in a predictable and consistent manner, a view that is supported by network theory (Borgatti & Halgin, Faraj & Johnson, 2011). The network exchange theory thus comprises both social and network exchange theories (Faraj & Johnson, 2011; Monge & Contractor 2003). Social exchange theory focuses on actions and interactions of individual developer's in the network and provides ways of studying collective outcomes while network exchange theory places

focus on network positioning on access to resources (relevant source code information) and power (ability to contribute to lines on leading source code contributors) including social capital (Monge & Contractor, 2003). A key driver in exchange theory is reciprocity where developers reciprocates source code modifications to the initiator or others (Flynn, 2005; Kilduff et al., 2006). Eheh (1974) noted that social exchange theory places high importance on reciprocity because developers are humans that keeps scores of actions on the source code modifications and change their subsequent digital practice actions based on perceived digital practice balance). Individual developer programmers are inherently social like any other developers and this study expects them to socially exchange information (i.e., intensify the level of participation in the modification of the source codes) in the technology based platform (Faraj & Johnson, 2011; Kling, 2003). This study therefore expects both the social and network exchange theories to come into play in the exchange patterns in digital platforms. On that bases, this study hypothesis that developers in digital platforms exhibit structural network tendency towards reciprocity. i.e.,

HYPOTHESIS 2: Developers in digital platforms contribute to codes in a manner that shows reciprocity to initial alteration of the source code

Another key characteristic in network formation model is closure or clustering. Closure or clustering can be explained from theories of mutual interest and collective action where developers tend to form ties, coalesce/cluster into groups because groups give them a collective ability to learn from other developers, and thus acquire gains that far outweighs those gained in individual code contribution (Marwell & Oliver, 1993, p.

2). We therefore expect developers in the nanoHUB.org to contribute to codes based on area of interest or expertise and this might lead to formation of mutual interest groups out of the digital practice activities. This study would therefore expect developers in online communities to participate in source code modification to specific set of source codes in a manner that clusters into groups with the hope that they tend to "gain" from engaging in "specializing" those group settings. This study therefore hypothesis that developers in online communities will contribute to source code modification in a manner that forms ties and coalesce/cluster into groups (clusters) that they believe will increase their collective ability to leverage and mobilize resources for collective action in the platform. i.e.,

HYPOTHESIS 3: Developers in digital platforms contribute to codes in a manner that show clustering patterns

4.3 Methodology

4.3.1 Data

The data for this study came from developer network of scientific digital platform (nanoHUB.org cyberinfrastructure) (NanoHUB.org, 2014). The network of developers was created through a developer's weight matrix described in the theoretical section. Our data from nanoHUB.org is organized by the structure of scientists that form the platform including data on tool developers, tool users, educators and leaners. The data for this study is comprised of 7-terms (of 6 months) panel data of developers that were available from the 2002, when nanoHUB.org was launched (Matei, 2004; nanoHUB.org, 2014).

4.3.2 Variables

The weight matrix was the main variable for this study because it was used to extract the network characteristics that were used to fit the model to data for the link to link and stochastic dominance models. The weight matrix was also used as the dependent variable for the ERG (p*) Model.

4.3.2.1 The Weight Matrix

We constructed the weight matrices for the 7 year panels following Matei et al. (2015) modelling of network formation from developer's level of digital practice (Please refer to Section. 3.2.1). The weights were used to construct the edge list and adjacent weight matrix of developers' collaboration in the network. The weight matrices for time slices 2 to 8 are presented in Figures 5 to 8 below.

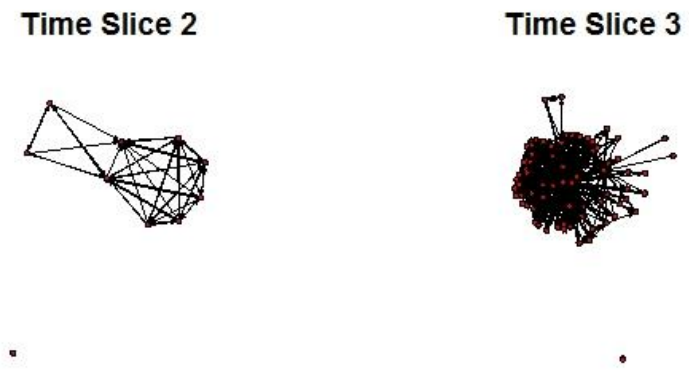


Figure 5: Developers Network at Time Slices 2 and 3

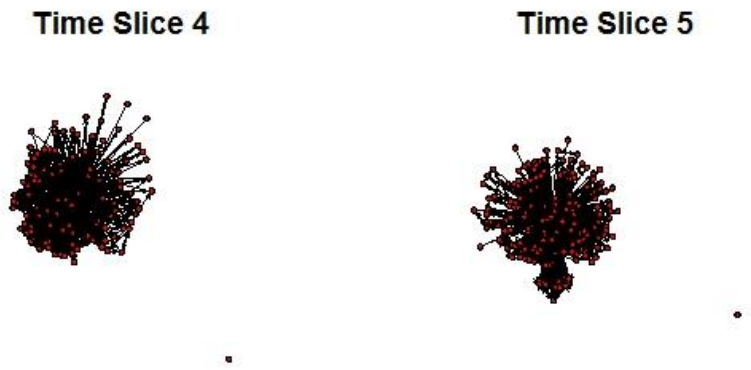


Figure 6: Developers Network at Time Slices 4 and 5

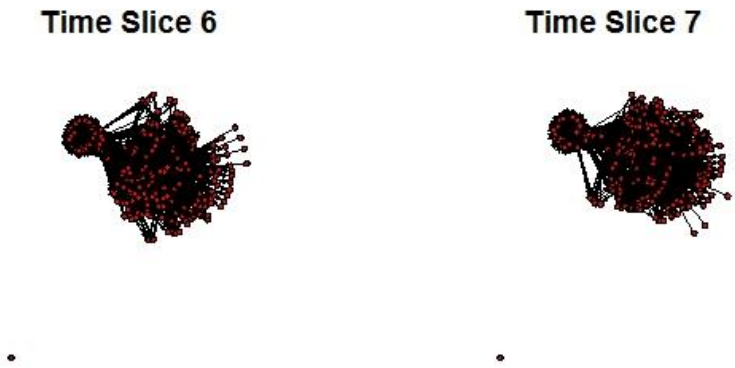


Figure 7: Developers Network at Time Slices 6 and 7

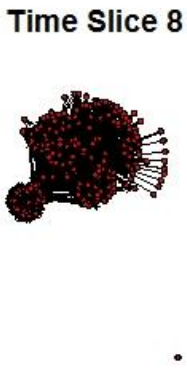


Figure 8: Developers Network at Time Slice 8

Figures 5 to 8 show gradual increase of developers from time period 2 to time period 8. The networks show different patterns implying that there are reorganizations taking place in the networks.

4.3.3 Model

Empirical analysis of socially generated networks have found the structures to exhibit five main characteristics: (1) nodes exhibit small average short path length between them, (2) clustering coefficient (tendency of linked nodes to have mutual neighbors) is high, (3) degree distribution tend to follow power law, (4) nodes tend to exhibit assortativity (degree of nodes tends to be correlated), and (5) clustering amongst neighbors, in some networks, tend to be inversely related to the node degree (Jackson & Rogers, 2007). The authors noted that the five characteristics are usually used to validate network formation models.

Formal modelling of network formation has generally followed two categories. The first category is strategic formation of network and this involves application of game theory tools while the other is more of the mechanical one and it describes the stochastic processes of network formation (this has its root in the random graph literature) (Amaral et al. 2000; Erdos & Renyi, 1959; Jackson & Rogers, 2007). These models lead to either scale free networks (networks that follow power law degree distribution) or uniform random networks (networks that follow a distribution that is negative exponential). The first random graph model was developed by Erdos and Renyi in 1959 (Erdos & Renyi, 1959; Lusher et al., 2013). The model states that every link is formed with probability $p \in (0,1)$ independent of any other link and it is mostly useful for understanding certain thresholds and how networks come to exhibit certain features (Jackson, 2008; Lusher et al., 2013). The model assumes that once the threshold is met the links will continue forming to one big component and this has been identified as a major caveat of the model because this seems to violate the above highlighted properties of social networks, e.g., clustering, degree distribution e.t.c (Jackson & Rogers, 2007). Improvements of Erdos Renyi (1959) model have involved modifying the model to capture those important network characteristics like clustering, degree distribution e.t.c. These include modelling network formation as uniform random graph and/or by preferential treatment (e.g., Barabasi & Albert, 1999; Cooper & Frieze; 2003; Watts, 1999). Recently, hybrid models have also been developed (e.g., Jackson & Rogers, 2007; Kumar et al., 2000; Vazquez, 2003). Other extensions include stochastic block modes, exponential random graphs models (ERGM) and newly introduced statistical exponential random graph models

(SERGMs) (e.g., Chandrasekhar & Jackson 2012; Chatterjee & Diaconis, 2011; Frank & Strauss, 1986; Lusher et al., 2013).

In this study we applied and compared results of two models while trying to explain developer network formation process in nanoHUB.org cyberinfrastructure. Our first model was a two stage process. In the first stage, we evaluated the network formation process by modelling our networks through a link-to-link ERG model following Jackson & Rogers (2007) hybrid model. In the second stage we identified the network formation characteristics of the most efficient network based on stochastic dominance criteria of degree distribution. i.e., we tried to trace back the network formation characteristics of the stochastically dominating network. In the second model, we applied Exponential Random Graph (p*) Model (ERGM). ERG(p*) Models are used to understand how and why social networks ties arise (Lusher, et al., 2012. p. 9; 16). An alternative model application of would have been the separable temporal ERGMs (STERGMs) that are an extension of ERGMs for modeling dynamic networks in discrete time (Krivitsky and Handcock, 2010). STERGMs consists of two models: one ERGM underlying relational formation, and a second one underlying relational dissolution (Krivitsky and Handcock, 2010). However, the link-to-link ERGM model and stochastic dominance criteria model was chosen over STERGMs because our networks in the seven time slices had unequal vertices (developers) (Krivitsky and Handcock, 2010). Moreover, as it will be described below, the link-to-link ERGM model applies the mean field theory that predicts the growth dynamics of the individual vertices, and is used to calculate the connectivity distribution and the scaling exponents (Barbasi et al., 1999; Jackson, 2008; Jackson & Rogers, 2007). The mean-field method was therefore used to address the

properties of two variants of the scale-free model, that do not display power-law scaling. Both models are described below.

Jackson and Rogers (2007) link-to-link ERG model is a simple network formation model that combine both random and preferential attachment formation techniques and was used to address the first hypothesis. Our model is also based on assumptions made about the features of digital developer network which follow similar pattern to coauthorship, citation and also worldwide web networks network-www (Albert et al., 1999; Gonzalez-Brambila et al., 2013; Jackson & Rogers, 2007; Li et al, 2013). Networks having these characteristics have been modelled through, random, preferential and hybrid models. Preferential models are given by power law which is linear but this assumption might be wrong as showed in www network where the distribution does not follow the law (Jackson & Rogers, 2007). Similarly, in citation network, people search for coauthors randomly then use preference to attach to others and therefore we cannot expect a pure power law distribution. We assume that our scientists are nonstrategic and the collaborations are a combination of uniformly random and preferential process (e.g., Jackson & Rogers, 2007). We proceed to describe the random and preferential models and finally the hybrid model. The models are adopted from Jackson (2008) and Jackson and Rogers (2007).

4.3.3.1 Random Model

Random model is based on both the graph and probability theories (Jackson, 2008; Jackson & Rogers, 2007). The model assumes that a new developer i uniformly randomly

picks and forms m links from a set of existing nodes (The average indegree is used as m-e.g., Jackson & Rogers (2007)). This gives developer's i starting condition degree as $d_i(i) = m$. The rate of change of degree distribution of scientist i is given by (23),

23) $$\frac{dd_i(t)}{dt} = \frac{m}{t}$$

Equation (24) is a differential equation which gives the following solution,

24) $$d_i(t) = m + mlog\left(\frac{t}{i}\right)$$

We can use equation (24) to get a degree distribution by solving for nodes that have expected degree of less than d at some time t, i.e.,

25) $$m(1 + log\left(\frac{t}{i}\right) < d$$

Solving for i gives the nodes that have expected degree of less than d are those born after, i.e.,

26) $$i > te^{-\left(\frac{d-m}{m}\right)} and\ \frac{i}{t} = e^{-\left(\frac{d-m}{m}\right)}$$

This gives a distribution function $F_t(d)$,

27) $$F_t(d) = 1 - e^{-\frac{d-m}{m}}$$

The distribution function (27) is a negative exponential.

4.3.3.2 Preferential Attachment Model

Preferential attachment model is based on the assumption that a new developer i picks m links from a set of existing nodes and forms m links based on probability that is proportional to their degrees (Jackson, 2008; Jackson & Rogers, 2007). The probability that an existing deevloper i gets a new link at time t is m times $i's$ over the total degree of all existing scientists at time t, i.e.,

28) $$\frac{dd_i(t)}{dt} = \frac{d_i(t)}{\sum_{j=1}^{t} d_j(t)}$$

But the total number of links in the system at time t is given as tm and $\sum_{j=1}^{t} d_j\,(t) = 2tm$ which changes (28) to,

29) $$\frac{dd_i(t)}{dt} = \frac{d_i(t)}{2t}$$

Solving the differential equation gives the distribution function,

30) $$F_t(d) = 1 - m^2 d^{-2}$$

And density/frequency distribution,

31) $$f_t(d) = 2m^2 d^{-3}$$

Which is a power law distribution.

4.3.3.3 Hybrid Model

The distributions of random and preferential models give two extremes distributions and hybrid models give an intermediate distribution where we assume that networks form through a combination of the two models (Jackson, 2008; Jackson & Rogers, 2007). This implies a scientist i forms a link m randomly by proportion parameter α and preferentially by proportion $(1 - \alpha)$. This gives the rate of change in the degree of a node as,

32) $$\frac{dd_i(t)}{dt} = \frac{\alpha m}{t} + \frac{(1-\alpha)d_i(t)}{2t}$$

Solving the differential equation using the steps above gives the indegree distribution function,

33) $$F_t(d) = 1 - \left(\frac{m+\frac{2\alpha m}{1-\alpha}}{d+\frac{2\alpha m}{1-\alpha}}\right)^{2/(1-\alpha)}$$

The distribution (33) follows power distribution (pure preferential distribution) when $\alpha = 0$ and near exponential distribution (random distribution) when $\alpha \to 1$. To solve for α we linearize and rearrange (33) to get

34) $$log\big(1 - F(d)\big) = \frac{2}{1-\alpha} log\left(m + \frac{2\alpha m}{1-\alpha}\right) - \frac{2}{1-\alpha} log\left(d + \frac{2\alpha m}{1-\alpha}\right)$$

Equation (34) can be written as,

35) $$log(y) = c - \beta \log\left(d + \frac{2\alpha m}{1-\alpha}\right)$$

Where

$$y = 1 - F(d)), c = \frac{2}{1-\alpha} log\left(m + \frac{2\alpha m}{1-\alpha}\right) \text{ and } \beta = \frac{2}{1-\alpha}$$

α in (35) will be solved through iterative process to determine ratio of random to preferential attachment.

4.3.3.4 Efficiency in Network Structure; Stochastic Dominance Model

We study the implications of the network formation process on the operation of a network through efficiency because we are evaluating developer network over a 7-year period. Efficiency of the model was tested by ordering the 7 degree distributions by first- and/or second-order stochastic dominance (Jackson, 2008; Jackson & Rogers, 2007). Stochastic dominance model was used to evaluate and distinguish the most efficient network structure. The dominance network structure was used to evaluate structural and operational characteristics that are important for formation and sustenance of developer network. i.e., the dominance network characteristics helped us tie the network formation characteristic to formation of developer network (Jackson & Rogers, 2007). The network formation characteristics that were used to validate the network formation model

included the average short path length, clustering, degree distribution pattern, assortativity and clustering and node degree relationship.

4.3.3.5 Exponential Random Graph Model.

ERG (p*) models try to describe the network statistics in the network so as to categorize/classify the network structure (Lusher, et al., 2013). The authors noted that ERGM model is not a social influence model but a “tie-based” model for social network i.e., models are not focused on predicting the outcome of individual in the network (e.g. diffusion or contagion) but it is about revealing patterns that may enable inferences on tie formation including social selection processes where network ties are predicted from the attributes of the network scientists (Lusher, et al., 2013). The ERGM model explains the “complex combination” of social processes that facilitate formation of network links (Lusher, et al., 2013). The authors noted that modelling ERGM requires the researcher to choose the set of statistics/configurations that he/she believes are theoretically sound for formations and/ sustenance of that particular set of network. The researcher then applies the model to an observed social networks and the parameters are estimated. This permits inferences about the type of social processes that are important in creating and sustaining the network (Lusher, et al., 2012). The authors noted that there are a whole set of ERGM models and the researcher chooses the specification of the ERGM for the data. ERGM is founded on muilti-theory process because of the complexity (multiplicity, interconnectedness and dependencies) of the network structures, configurations and processes (Lusher, et al., 2012. p. 10). One main theory of ERGM theory is interdependencies of ties and ERGM can test the evidence as to which processes

contribute to the formation of the network structure (Lusher, et al., 2012. p.21; Monge & Contractor, 2003)

The main network theories that we investigated with ERGM model are reciprocity or exchange and this is configured from dyadic process. Other relationship involving triads deal with mostly clustering and closure (path or network closure). Out-2 star is a star like structure with two outgoing ties from the central node and this is used to denote activity-based configurations, where an actor directs ties to many network partners (Lusher et al., 2013). The opposite of that is in-star configurations and these measure popularity of an actor. i.e., an actor has two incoming ties. The other configuration represent homophily i.e. actors of the same attribute have reciprocated ties. A general ERGM model with edges, stars and triangles is given by,

36) $$p(Y = y_{ij}) = \frac{1}{Z} e^{(\theta L(y) + \sum_{r=2,n-1} \sigma_r S_r(y) + \tau T(y))}$$

Where,

Y is the software developer adjacency matrix

y_{ij} is binary indicator for edge (i, j)

$L(y)$ is the number of edges

θ is the edges or density parameter to be estimates and

S_r is the number of stars of size r in y

σ_r is a parameter for a star of size r

τ is clustering or triangle parameter

$T(y)$ is the number of triangles

Z is the normalizing constant which is a function of parameter vector and this ensures that (15) is a probability distribution.

All ERGM models take the above general form, describing the probability distribution of graph on nodes.

4.4 Results and Discussion

4.4.1 Mean Field Method and Network Characteristics

Mean filed approximation was used to fit the data to the degree distribution based on the dynamic hybrid model (equation 33[16]). This method was used to establish the network formation characteristics based on the variance of α. The variance of α gives us an indication of preferential to uniform random attachment which tells us how the links are formed in nanoHUB.org developers network (Jackson, 2008; Pennock et al., 2002). m, the number of new links formed in each period was directly calculated from the data. m is calculated as half of the average degree (Jackson, 2008). Table 8 shows m to range from 3.5 to slightly above 30. The proportion of uniformly random connection in developer network (α), was then calculated through a simple iterative least square approach. The simple iterative least square approach starts with an initial guess of α, $e.g., (\alpha_0)$. Equation (35) is then regressed with (α_0) in place to get the parameter estimate β that is used to calculate a new (α_1). α_1 is used as the" new guess" and the entire regression is repeated to calculate a new α. The iteration process continues until

[16] The link to link hybrid model is given by $log(y) = c - \beta \log\left(d + \frac{2\alpha m}{1-\alpha}\right)$

the initial and estimated α converge. Results of estimated α are presented in Table 8 below.

Table 8: Link to Link Network Statistics and Developer Network Characteristics

	Time						
	T02	T03	T04	T05	T06	T07	T08
Number of nodes	289	297	294	179	108	74	10
average in-degree: m	31.8	31.03	31.09	11.014	9.13	10	4.5
α-Proportion of	0.998	0.999	0.999	0.999	0.998	0.999	0.998
Diameter data	8, 241	17, 35	2,232	3,62	3,72	3,52	2,7
Average Short path	2.156	2.162	2.156	2.2315	1.995	1.937	1.484
Assortivity	9.0e-3	6.54e-3	1.23e-2	4.13e-2	9.e-3	6.6e-3	9.8e-2
Cluster Coeff	0.897	0.894	0.9	0.328	0.35	0.489	0.854
Dyads (mutual)	1247	1246	1219	528	214	155	13
Density	0.1089	0.103	0.104	0.0591	0.081	0.13	0.444
Reciprocity	0.2753	0.275	0.271	0.5614	0.46	0.442	0.667

Results in Table 8 show α in all time periods to be close to 1. When $\alpha \to 1$ it approaches exponential distribution even though the limit is harder to get (Jackson, 2008). Approaches to fitting such data include Berger et al. (2005) polya urn models and simulations for degree distribution using non-parametric bootstrap techniques (Jackson, 2008). We fitted our degree distribution model results to data using nonparametric bootstrap technique The Nonparametric bootstrap method takes the original data set as the population and then draws equal samples by simulation. The sampling is done through replacement method that ensures that each observation has the same probability of being picked. Table 9 shows the nonparametric bootstrap results of the relative

frequency of the degree distribution in time slice 2 (The actual plot of the degree distribution is presented in the Appendix). The first column shows the important distribution statistics that were tested in the by bootstrap model; Mean, variance and Median. The second column gives the original data estimates of the statistics while the third shows the bias i.e., the difference between population value of the degree distribution and the expected value of the link to link degree distribution. The fourth and fifth columns show the standard error and the percentile (lower and upper confidence intervals) of the bootstrap estimates.

Table 9: Nonparametric Bootstrap Estimates for Fitting Degree Distribution

Statistics	Original (t*)	bias	Std. error	Percentile
Mean	-4.973	-0.002	0.095	-4.994, -4.603
Variance	0.766	-0.012	0.093	0.577, 0.939
Median	-4.973	0.011	0.134	-4.973, -4.568

Results show that the mean and median estimates of the relative frequency of degree distribution are similar, -4.97 while the variance of the mean is 0.77. Results show that the difference between mean, variance and median population relative frequency values (bootstrap) of the degree distribution and the link to link degree distribution value to be low (low bias). Results also show that for a 95% confidence interval, we find the 2.5%-tile and 97.5%-tile mean and median relative degrees frequency in the distribution to be -4.99 and -4.60 and -4.97 and -4.57 respectively. These results imply that we are 95% confident the degree distribution from our link to link model fits the data. The results imply that developer networks do not follow a network-based coding pattern but chose codes to work on in almost uniform randomly manner. While we would expect a more

predominantly network based pattern of coding in developers' network of nanoHUB.org cyberinfrastructure, results imply that new developers contribute to source codes in a near uniformly random manner. While we would expect a more predominantly network based pattern of meeting in developers' network of the nanoHUB.org cyberinfrastructure, results imply that new developers attach to existing developers in a near uniformly random manner. These results are not surprising, because, contrary to expectation and earlier studies about formation (e.g., Barabasi & Albert, 1999; Faraj & Johnson, 2011; Jackson & Rogers, 2007), online enabled networks links form in a manner that does not follow preferential attachment. To see a clear distribution pattern of the degree distribution, we plotted scatter plots of the log of frequency against the log of degree. The plots are shown in the Appendix C excluding the one for time slice 8 that had very small data points. The figures show a similar pattern that follows negative exponential which is characteristics of uniformly random formed distributions. The tails are however fat like those of power law fitted distributions.

The link to link mean field approximation can also be used to fit for other network features such as small world (short diameter), clustering and assortativity (Jackson, 2008; Jackson & Rogers, 2007). Given that our model fit results gave us a 95% confident the degree distribution from our link to link model fits the data we expect other networks statistics that are derived from the network to follow; these statistics are validated by the ERGM model, nevertheless. For example, Jackson and Rogers (2007) fitted six different network model results to data and found near match. The authors found clustering from data to match the model fit in 3 of the 5 networks and found diameter of the data to be within the model fit in 5 networks. As such we will calculate network characteristics

from the data and establish the distinguishing network characteristics responsible for formation of nanoHUB.org developer's network. However, given that we are considering 7 time periods that have varying characteristics, we identify the most efficient network in terms of degree distribution and use its network characteristics to map out the distinguishing characteristics that are responsible for network formation. The identified network characteristics were further validated with the ERG(p*) model. Efficiency of 7-degree distribution was evaluated by ordering the distributions by stochastic dominance criteria using KS-test (Kolmogorov-Smirnov) (e.g., Jackson, 2008; Jackson & Rogers, 2007).

4.4.2 KS Efficiency Tests for Stochastic Dominance

KS-test is a non-parametric method that is used to evaluate whether two distributions differ significantly (Scaillet & Topaloglou, 2010). Table 10 shows the KS-test results for stochastic dominance. The first column shows the distribution that is to be evaluated (treatment distribution) against the reference distribution (control distribution) in the second column. Results show that distribution of time slice 2, 3 and 4 stochastically dominates those of time slice 6-8. Results also show that the distribution in time slice 5 stochastically dominate the one of time slice 8. Given that the time slice 2, 3 and 4 dominate those of the later periods (6, 7 &8) we will use and compare the distribution of the latest time period (time slice 2) and compare it with the latest time distribution amongst the dominated distributions (time slice 6). The alternative would be to compare the mean and standard deviation of the dominating (2, 3 and 4) and dominated distributions (6, 7 and 8) but this will imply that we are assuming the distributions are

normally distributions which might not the case. Therefore, we chose time slices 2 and 6 since those have more data points than those of 3 and 4 and 7 and 8 respectively. The dominated distribution time slice 6 was taken as the control while the dominating distribution, time slice 2, was taken as the treatment.

Table 10: KS Efficiency Tests for Stochastic Dominance

Time Slice 1 (T1)	**Time Slice 2 (T2)**	**Difference (T1-T2) in Distributions**	**p-value**
Time slice 02	3	0.0909	0.100
	4	0.0882	0.119
	5	0.0854	0.142
	6	0.124**	0.008
	7	0.119**	0.012
	8	0.201***	0.000
Time slice 03	4	0.074	0.268
	5	0.0799	0.197
	6	0.119**	0.012
	7	0.113**	0.019
	8	0.196***	0.000
Time Slice 4	5	0.069	0.356
	6	0.107**	0.030
	7	0.102**	0.046
	8	0.185***	0.000
Time Slice 5	6	0.039	0.950
	7	0.039	0.950
	8	0.116**	0.016
Time Slice 6	7	0.041	0.916
	8	0.077	0.230
Time Slice 7	8	0.083	0.168

***,** denote significance at 1% and 5% significance level respectively

The network of the dominating (treatment) distribution (time slice 2) was used to tie the network formation characteristic to the network outcomes when compared to the

dominated one. That is, we used the direction and magnitude of the network formation characteristics including average short path length, clustering, degree distribution pattern, assortativity and clustering and node degree to validate the network formation characteristics of developer network in nanoHUB.org cyberinfrastructure (e.g., Jackson & Rogers, 2007). The network characteristic results are presented in Table 8.

Results in Table 8 show that assortativity coefficients of networks in both time slices to be approximately -0.009. The low assortativity coefficient imply that developer networks are characterized by low degree homophily (Newman, 2003). The clustering coefficient for the more efficient (treatment) network, time slice 2, is about 0.9 while the one for the inefficient (control) time slice 6 is about 0.35. Clustering coefficient is also known as transitivity coefficient and a higher coefficient value in time slice 2 over 6 imply that there is a high number of triangles, transitive closures in developer network (Gabor, 2014; Wasserman & Faust, 1994). The measure for reciprocity is usually given by reciprocity coefficient or density. Table 8 shows that the reciprocity for the control network (time slice 6) is higher than the reciprocity for the treatment network (time slice 2), compare reciprocity coefficient 0.46 to 0.27 for time slice 6 to time slice 2 respectively. These results imply that developer networks are characterized by low reciprocity. Reciprocity defines the proportion of mutual connections, in a directed graph. i.e., the probability that the reverse link of a directed edge is also featuring in the network. (Gabor, 2013). Result imply that about there is low probability (about 27%) of mutual connections or social exchange between developers in the network. The density of treatment network (time slice 2) is low and insignificantly different from the control network (time slice 6); compare 0.11 for treatment to 0.08 for control in Table 1. Low

densities imply that developer network are characterized by a relatively low number of mutual ties than they would be in a purely random network.

4.4.3 ERGM Model Results

We use ERGM (p*) to further validate the presence (and absence) of network ties, and so provide a model for developer network structure (Lusher, et al., 2013). ERGM is a "tie-based" model for social network and allows us to understand the "complex combination" of social processes by which network ties are formed (Lusher, et al., 2013). In modelling ERGM we were guided by the findings of the above highlighted network characteristics and configurations that we believe are responsible for the formation and sustenance of developer network in digital platforms. We therefore tested the presence or absence of reciprocity, clustering, assortativity, diameter and tendency to attachment in a uniform randomly in developer network using ERGM (p*) model. The network statistics that we considered to test for reciprocity, clustering, assortativity, and non-preferential attachment were mutual dyads, triangles/transitive/cycles, gwdidegree, and istar respectively (e.g. O'Malley & Marsden, 2008). However, because of the computational complexity nature of ERGM (p*) model, whereby, for example, inclusion of both istar and triangles leads to model degeneracy and lack of convergence, we did stepwise and near permutation combination[17] of the variables so as to best capture the magnitude and direction of the variables (e.g., Hunter et al., 2008; Lusher et al., 2013; O'Malley & Marsden, 2008).

[17] All model combinations did not alter the direction of the network statistics.

We tried various variable combinations in 4 models to evaluate their composition in developer network. Models 1 to 4 represent different combinations of the desired network characteristics. Model 1 has only mutual ties while model 2 has both mutual and transitive network statistics. Model 3 has mutual, transitive and istar (3) network statistics and model 4 has mutual and gwidegree network statistics. As aforementioned, mutual statistics is used to evaluate for presence of reciprocated ties in the network (Lusher et al., 2013; O'Malley & Marsden, 2008). Transitive triad tests for presence of clusters in the network, istar (3) network statistics test for the presence of preferential attachment while gwidegree (t-2.5) test for the presence of degree homophily in the network (e.g., Lusher et al., 2013; O'Malley & Marsden, 2008). All the models converge and estimates results are presented in Table 11.

Table 11: ERG (p*) Model Results

	Model 1	Model 2	Model 3	Model 4
Variable	Estimate (std err)			
Mutual	-2.519*** (0.030)	-13.534*** (2.003)	-2.675*** (0.107)	-0.926*** (0.026)
Transitive		0.024*** (0.001)	0.002*** (0.000)	
Istar (3)			-3.316e-04*** (3.677e-06)	
Gwidegree ($\tau = 2.5$)				-12.938 (369)
AIC	74635	92529	74135	56699
BIC	74644	92548	74163	56717

*** denote significance at 1% significance level

Results show mutual ties is negative and statistically significant in all the four models while istar (3) is negatively significant in model 3. This implies that the developer network does not to reciprocate nor follow preferential attachment while forming. i.e.,

developers do not reciprocate codes and do not follow a particular preference when joining the network. The findings are well supported by those of stochastically dominating networks found in the link to link model. The link to link model showed that developer network tends to be characterized by low reciprocity and forms in a manner that follows uniformly random pattern. Both model results imply that we reject the hypothesis that developer network is highly characterized by reciprocated ties. Results also show that we uphold the hypothesis that developer ties form in a manner that follows uniform random attachment. Table 11 shows that transitive triad's statistics to be positive and statistically significant in both models 2 and 3. These result imply that developer network form clusters (exhibit closure) than they would in a network that is formed in a pure uniform random manner. i.e., these results are also supported by our network link to link results that showed that the dominating network showed tendencies for high clustering. We therefore uphold our third hypothesis which posited that software developers in the nanoHUB.org are characterized by high clustering.

4.4.4 Goodness of Fit of the Models

We further subjected the model to goodness of fit (GoF). The graphical tests of GoF are presented in Figure 7. The graphical tests of GoF technique is chosen over the traditional AIC, BIC and likelihood methods because the plots are more informative than the AIC or BIC for they tell us which structural features fit well and which do not (Hunter et al., 2008). Moreover, the GoF plots does not rely on the assumptions that observations need to come from an independent and identically distributed sample which is a requirement for calculating AIC and BIC (Hunter et al., 2008). The authors also noted that likelihood ratio method is only applicable to dyads independent and not dependent models like

ERGM (Hunter et al., 2008). Gof compares the set of observed network statistics with a range of the same statistics obtained by 100 simulations of networks from the fitted ERGM (Hunter et al., 2008). We fit our model using three commonly and important network statistics including degree, shared partner statistics and geodesic distance. Hunter et al. (2008) pointed out that degree statistics gives an indication of the distribution, while shared partner statistics gives an indication of triangle count because triangles are a function of shared partner statistics. The authors then noted that geodesic distance gives a basis of the two most common features, centrality and are also important to understanding the speed and robustness of transmission. GoF computes the p-value for the geodesic distance, degree and average short path summaries to ascertain the ERGM models goodness-of-fit. The GoF graphs for the four models are presented in Figure 9 below.

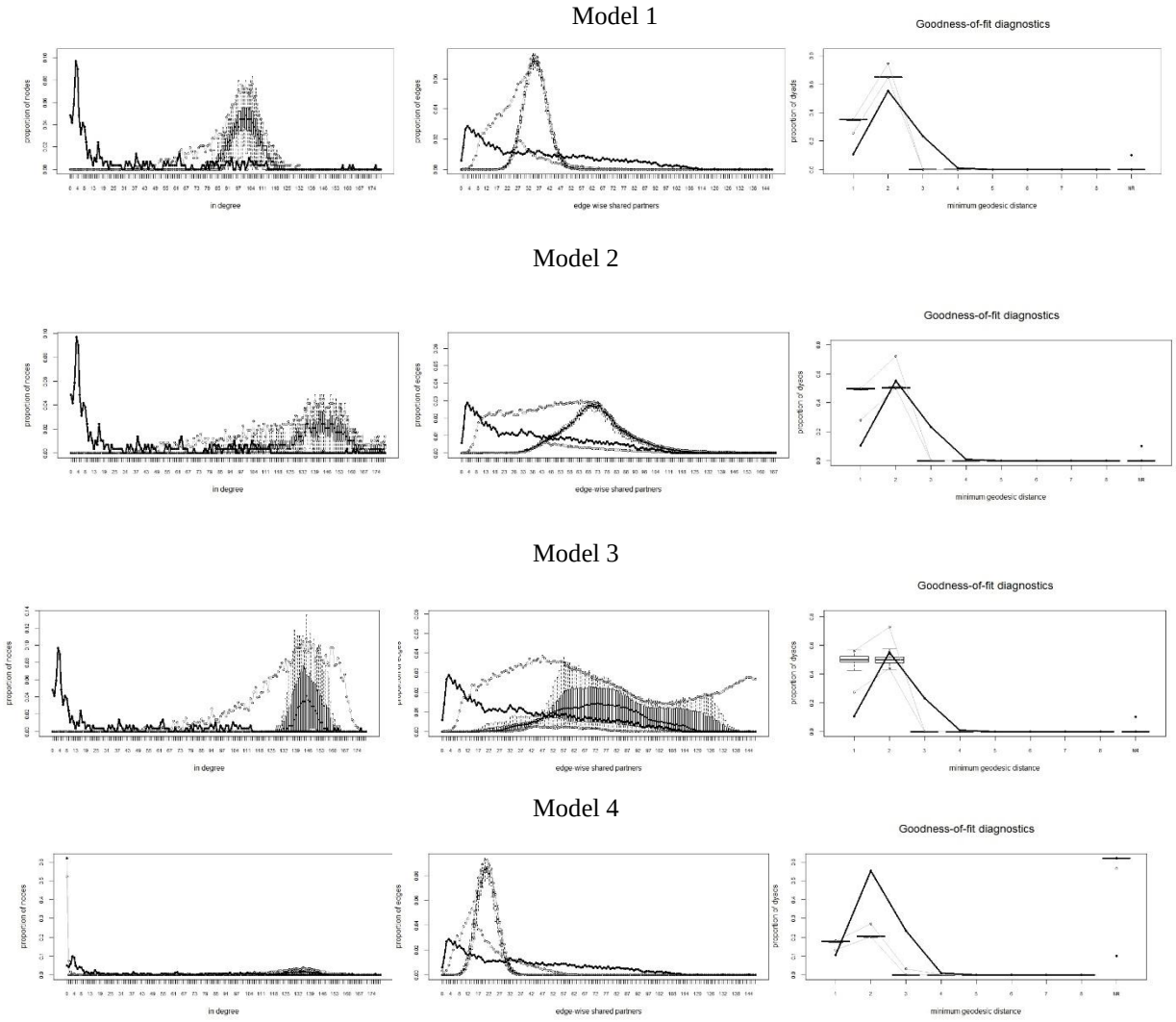


Figure 9: Simulation Results for Dyadic Dependence ERGMs of Table 12. (Model 1) Mutual. (Model 2) Mutual + Transitive. (Model 3) Mutual + istar (3) + Transitive. (Model 4) Mutual + gwd ($\tau = 2.5$).

Figure 9 show results of 100 simulations for developer network from fitted dyadic independence models given in Table 11. Columns one to three show the fitted network statistics (degree, shared partner statistics and geodesic distance) for the four models. The vertical axis is the log-odd ratio of the relative frequency, and the solid line is the statistics of the observed network. Results show that the four models have different fits

for all the three network statistics implying that the models have varying magnitudes of either strongly estimating and underestimating the degree distribution, local clustering and average short paths of the network. Figure 9 show that model 4 fits better than models 1-3 for indegree and edgewise-shared partner while model 1 fits the geodesic distance better than the others. These results imply that model 4 is best suited for estimating attachment patterns (preferential to random), reciprocity and clustering while model 1 best for estimating the network average short distance which is not evaluated in this study nonetheless. The results give a relatively fair representation of the data fit. Gof allows us to know whether the specified model for our observed data represent particular network structures of graph features well but should not be expected to explain or fit all features of a network (Lusher et al., 2013).

4.5 Summary and Concluding Remarks

This study sought to understand the network formation, operation and organization (collaboration) and sustenance mechanism in an online enabled cyberinfrastructure (nanoHUB.org) through social network modelling. A simple link to link network formation model was used to evaluate the network formation pattern. Stochastic dominance model was used to evaluate the most efficient model which was used to evaluate and fit the network characteristics that are important for developer networks. ERG (P*) model was used to compare and validate the network formation characteristics of the developer network. The study was anchored in theory of network that mostly explains the patters of the network formation. Other network self-organizing and sustenance sub-theories including tendency for the networks to show reciprocity and

clustering were also tested in the model. Both link to link and ERGM models results show that developers contribute to source code in a manner that follows a pure uniform random distribution. These results confirm our hypothesis that online communities form in manner that inclines more towards a pure random attachment and are similar to those found by other online studies (e.g., Faraj and Johnson, 2011; Jackson, 2008). The practical implication of this study finding is that online platform managers should put least efforts in activities that try to influence developer's involvement in community activities. Results also showed that developer are characterized by low tendencies to reciprocate but have a high tendency to form clusters. These results imply that developer's participation in online communities is not influenced by back and forth exchanges of code modifications e.t.c. but flows exchanges that tend to coalesce (cluster) in small groups naturally. These results imply that platform managers should put least efforts in activities that enhance to direct exchanges in the SVN files. Results have also shown that developers are characterized by low homophily, that is, developer network exhibits heterogeneous coders working on a particular tool. The theoretical contributions of this study are: (1), application of different ERGM models to understand the network formation and organization patterns of online developer community that includes formation in a pattern that follow pure random distribution, network exhibiting low reciprocity and homophily but high tendencies to cluster and; (2), application of stochastic dominance to order most efficient distribution in terms of degree distribution.

CHAPTER 5. COMMUNICATION CHANNELS AND SOCIAL STRUCTURES ASPECTS OF DIFUSSION OF SOFTWARE IN ONLINE DIGITAL USER COMMUNITY: A BASS MODEL AND NETWORK AUTOCORRELATIVE MICRO MODELLING

5.1 Introduction

Diffusion of innovation studies have broadly focused on timing, innovation, communication channels and social structures aspects of transmission after Bass model of diffusion was introduced in marketing in the 1960s (Bass, 1969; Mahajan et al., 1990; Rogers, 1983). Bass (1969) claimed that diffusion patterns are the product of interaction between innovators (early adopters) and imitators. In more terms, "The basic assumption of the model is that the timing of a consumer's initial adoption of a scientific artifact is related to the number of previous adopters". Bass model is therefore a summative model that describes diffusion in terms of the behavior of the entire user network; the model largely ignores the social systems on which the innovation impacts (network structure) (Bass, 1969). Bass model is based on the assumption that users are fully connected (in fully connected component) and are homogeneous which implies that every individual has some possibility of influencing the other through the network. i.e., there is social contagion due to homogeneity in the social networks. Bass model is therefore good at looking at the timing[18] aspects of diffusion of innovation but not the social structures and

[18] We use timing to denote all aspects of adoption curve including speed and the saturation

communication channel aspects of innovation Bulte & Stremersch, 2004; Laciana et al., 2013). Bulte and Stremersch (2004) and Peres et al. (2013) pointed out that the model does not provide an insight about the processes that determine adoption, or how individual's social interactions are linked to the global social behavior because of the assumption of complete network connectedness and social contagion which might not be being realistic in real world. The authors continued to note that that diffusion process (i.e., the typical logistic-S-Shaped diffusion curve[19]) does not essentially come from social contagion process but due to some intrinsic tendency of heterogeneous individuals to adopt and this is better explained by microscopic models. Matei (2014) argued that different structures and patterns of user network are largely determined by the level of interactions in digital practice space.

Microscopic (Or Micro) models are commonly referred as agent based models because they evaluate individual's (agent's) behavior including the innovation characteristics and social interactions that influence adoption (Fibich & Gibori, 2010; Laciana et al., 2013). The models relate explanatory variables (covariates) to adoption behavior and are therefore able to look at the social structures and communication aspects to the diffusion process and therefore overcome the some of the limitations of the macro based models including homogeneity of the users (Jackson & Rogers, 2007; Meade & Islam, 2006; Rogers, 1983). This study will refer to the social structures and communication channels broadly as digital practice variables that have direct influence on an individual's tool adoption choice. The structural features of network have direct

[19] S-Shaped diffusion curve is similar to logistic function or normal function with heavier tails

influence on information flow that enhances diffusion of tools or technologies but the mechanisms and processes of communication which influence diffusion processes in the established network is limited (Jackson, 2008, p. 178).

Research on diffusion of innovation has largely focused on the macroscopic or microscopic perspective or a combination of both models (Laciana et al., 2013; Meade & Islam, 2006). The models choice and their effect on understanding the above highlighted four drivers of innovation (timing, innovation, communication channels and social structures) is contradictory and not very well understood (e.g., Laciana et al., 2013; Meade & Islam, 2006). Moreover, majority of these studies have applied simulation and analytical techniques with very little empirical evidence to buttress their findings (Ballester et al., 2006; Banerjee et al., 2013; Kitsak et al., 2010; Meade & Islam, 2006; van Eck et al, 2011). There are few empirical studies that have looked at the effect of network on diffusion (e.g., Ballester et al., 2006; Banerjee et al., 2013; Meade & Islam, 2006), and no study (to the best of our knowledge) has looked at diffusion from a network autocorrelation perspective in a non-market based digital platform. This study is a first (to the best of our knowledge) empirical application of diffusion model in a non-market digital user community using both the macro and micro diffusion models (e.g., Banerjee et al., 2013; Peres et al., 2010; Shanahan et al., 2008). The study findings contribute to the literature of understanding of the information flow from the network characteristics perspective and its impact enabling diffusion of tool diffusion in a non-market based online community.

5.2 Theoretical Framework and Hypothesis

Diffusion of innovation falls under diffusion theory which is a theory of communication[20] (contagion). The theory seeks to explain how a new product, practice or innovation (including diseases, computer virus) spreads amongst people that are interconnected through a network structure (Jackson, 2008.p.185; Mahajan et al., 1990). Diffusion theory is therefore nested in network theory that explains the effect of a network on productivity or choice (Borgatti & Halgin, 2011). The network structural characteristics (conduits of communication amongst people) facilitates flow of information in the interconnected structure (network) through a pattern that closely translates to product life cycle or the adoption curve (Borgatti & Halgin, 2011; Rogers, 1983). The product life cycle or the adoption curve shows the stages that a new product or innovation goes through while cascading through the social structure (network) and this process follows a distribution that is logistic or near normal (Rogers, 2003). Rogers (2003) categorized the adoption curve into the 'popular' five phases; innovators, early adopters, early majority, late majority and laggards (Mahajan et al., 1990; Rogers, 1983). The authors noted that peoples perceived ratio of benefits to cost (BCR) is a big factor that determines the speed and rate of adoption of innovation.i.e., an individual choice of adopting an innovation is directly related to their perceived benefits of adopting against not adopting. Some factors that are said to increase/alter BCR include modernity, homophily, physical distance and characteristics of opinion leaders all of which reduce the perceived risks and the initial effort required to learn about a new product before uptake (Jackson & Rogers, 2007;

[20] Communication is defined as any means that enable information sharing (Rogers, 1983)

Mahajan et al., 1990). Modernity refers to individuals going with the current social trends in the society while homophily is the tendency of individuals to associate with similar others (Jackson & Rogers, 2007). The authors further noted that physical distance is the space between two individuals which has direct influence on the speed of information flows between them while characteristics of opinion leaders refers to the direct influence of the leaders on information spreading and decision making. Mahajan et al. (1990) broadly classified these factors as external and internal and defined external factors as shocks from mass media (advertisements) and the internal factors as interpersonal communication within the network structure. The innovators and early adopters are said to be part of the visionary minority who experiment and take up an innovation or new ideas; these persons are also very entrepreneur and often risk takers (Bulte & lilian, 2001). On the other hand, the late majority are the skeptical mass who are risk averse because they wait until other individual take up an innovation/product before adopting an innovation/product (Mahajan et al., 1990; Rogers, 1983). A schematic representation of the adoption curve with the five phases of adoption is presented in Figure 10 below.

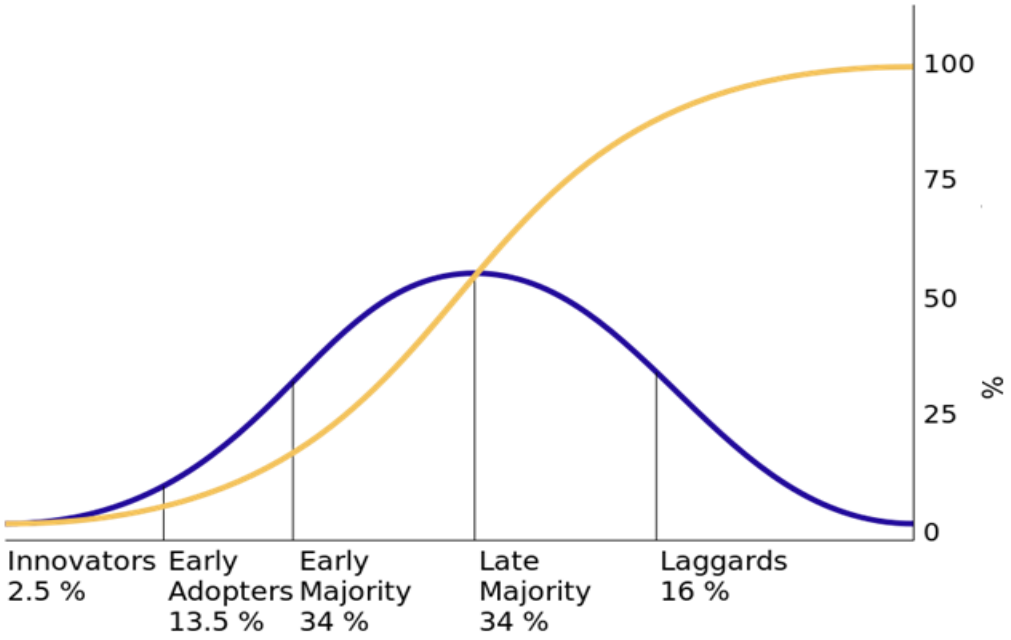


Figure 10: Adoption Curve Showing 5-Phases of Adoption (Rogers, 1983)

Diffusion process in online communities follows similar patterns with those of markets goods because it is the connections or channels of communication that influence the process (Firth et al., 2006; Susarla et al., 2012). Like other market-based connections, online communities involve users' scientists that are linked online with computers and their relationship is enhanced through digital practice (Matei, 2014).

This study used data from tools users in the nanoHUB.org to create online user community. The nanoHUB.org cyberinfrastructure is freely accessible for use by anyone with a nanoHUB.org account and was used to study the patterns of diffusion amongst its users (Klimeick, 2008; McLennan, 2012). The nanoHUB account registration is free but requires a user to have Java 1.4 or more, enable Javascript and cookies (McLennan, 2012). The web registration information and the cookies are used to capture usage and user information. The usage rate and reviews of a tool are all publicly displayed in the

nanoHUB.org cyberinfrastructure.org and these features plus the other tools specifics features including title, purpose, developer's name/s, program launch, class schedules largely constitute the internal and external influences that determine the rate and speed of adoption of individual tools (McLennan, 2012). Tool users therefore rely on these two distinct mechanisms when making a decision of whether to adopt a tool or not i.e., innovators are those scientists that go to the website to try a tool based on the initial appearance and other features on the website or based on advertisement flyers, titles e.t.c. and imitators are the tool users that rely on reviews, trusted user developers, other users direct or indirect influence. We therefore expect innovators to be impervious to the above highlighted network related influences when adopting a tool and the remaining users to be impressionable to internal /social influences (, i.e., modernity, homophily, physical distance and opinion leadership (e.g., Bulte & Lilian, 2001; Wright et al., 1997).

We model the above highlighted internal features as network proximity features and characteristics that increase the influence how information cascades amongst users in the network. Our reasoning is that digital practice activities are enhanced by both computer and location proximity features. We created a probabilistic proximity index weighted adjacency matrix based on nearest user using digital and geo-locational proximity features. Other studies have constructed weight matrices based on mostly geo-locational and online interaction proximity index such as gravity model and social models such as friendship, interaction, latent and following graph models (e.g., Leenders, 2002;(Jin, Chen, Wang, Hui, & Vasilakos, 2013) Matei et al., 2015; Winfree et al., 2005). The digital connections feature that we considered included internet protocol (IP) address, IP domain (Media Access Control-MAC), IP city, IP region, and IP country

while the geolocation features that we considered included the city, state and country. While the physical locations may seem as duplicates we hypothesize that there might be a low probability that people meet physically while doing their daily chores and influence each other. The digital proximity was given higher weight than the geo-locational weights nevertheless. The order of weighs was also tilted to favor those who share IP address, IP domain, IP city, IP region and IP country in that order based on the assumption that those the level of digital practice activities diminish in similar fashion. As such we created a probabilistic proximity index that sums to 10 based on an intuitive sense of the likely scenarios of interaction or encounter (We believe that this index can be improved based on some historical data).

Table 12: Proximity Index Scores for Adjacency matrix.

Digital proximity variables	**Score**
IP address	4
IP domain	3
IP city	1
IP region	0.6
IP country	0.4
Geo-locational Proximity Variables	**Score**
City	0.5
State	0.3
Country	0.2
Total	**10**

Users that share ip address and domain were given a high score of 4 and 3 because we believe that these have direct influence on each other's work hence have more propensity to contagion through both physical and information sharing (digital practice). Users sharing IP city, IP region and IP country were given low scores of 1, 0.6 and 0.4

respectively because we believe the probability of interacting and influencing each other diminishes based on the size of location that people might meet. Each proximity score was used to create separate adjacency matrix such that two users i and j will be connected by a weight 4/10 if they share the same IP address. Similarly, two users i and j will be connected by a weight 3/10 if they share the same IP domain (MAC). All the adjacency matrices were added to come up with the proximity index. Figure 11 shows the resulting user network (left) and the largest component (right).

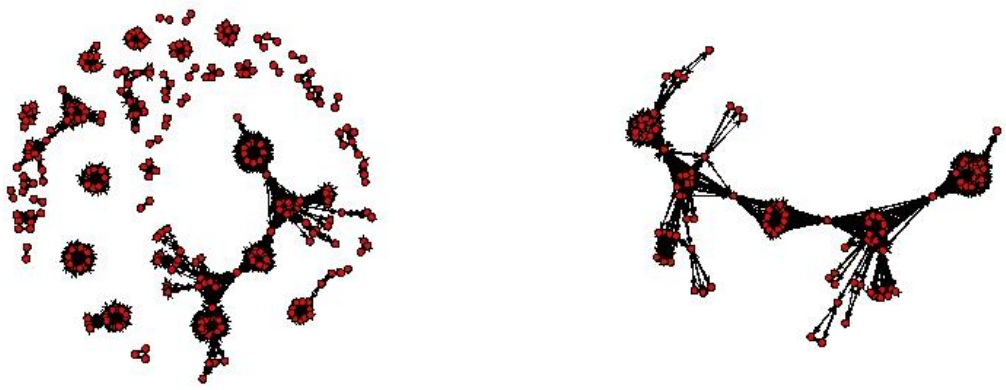


Figure 11: Spring 2006 the nanoHUB.org User Network and Largest Component

The resulting network shows the network with several components. The largest component (plotted to the right) was extracted and used to evaluate diffusion in that network because diffusion occurs in interconnected network structure (Jackson, 2008).

The network centrality measures that we consider as being significant to the diffusion process and also representative to modernity, homophily, physical distance and

opinion leaders include degree, closeness, betweeness, eigen vector centrality amongst others (Jackson, 2008; Gonzalez-Brambila et al., 2013; Li et al., 2013; Valente et al., 2008). Degree centrality measures the number of direct ties that a user is linked to in the network (Valente et al., 2008). Jackson (2008) noted that the number of direct ties facilitate combination and direct exchange of information that will increase the probability of a user adopting a tool. We therefore hypothesis that users with high degree centrality have a higher chance of getting information that will influence their decision to adopting a tool.

Hypothesis 1: High Degree centrality will be positively correlated with adoption of scientific innovation

Closeness centrality measures the average distance of a user to all others users in the network while betweeness centrality measures user's relative position in spanning the structural hole (Jackson, 2008; Valente et al., 2008). The two centrality measures are closely related to density of the network because dense network will lead to high measure of closeness but low betweeness centrality and vice-versa. Coleman (1998), Hansen (1999), Obstfeld (2005), Uzzi (1997) and Valente et al. (2008) argued that more dense networks facilitates direct access to information because users share norms in behavior and develop trust that they could use to mimic their fellow users and therefore adopt a tool i.e., closeness centrality will give a user a higher probability of adopting a tool because they are able to transfer tacit information based on their proximity and this will enable them make a decision to adopt or not adopt a tool. An opposing view point is that by Burt (1992, 2004) and Hargadon (2002) who argued that such information becomes redundant after sometime and that users in less dense networks are likely to gather

information that generate leverage to constructing an efficient and information-rich network where redundant partners is minimized. i.e., betweeness centrality will give users a higher probability of adopting a tool because structural holes facilitate diffusion of tools. Following these constructing views we will hypothesize the two centrality measures to take any but opposite directions in the digital platform.

Hypothesis 2: Closeness Centrality will be positively correlated with adoption of scientific innovations and Betweeness Centrality will be negatively correlated with adoption of scientific innovations

Hypothesis 3: Betweeness Centrality will be positively correlated with adoption of scientific innovations and Closeness Centrality will be negative correlated with adoption of scientific innovations

Eigenvector centrality measures the users relative position to opinion leaders (well-connected users). It is hypothesized that a user association or connection with opinion leaders will enable him have good contacts and information about a tool and this will increase his probability of adopting a tool. We therefore hypothesize that eigen vector centrality will increase the probability of tool adoptions. i.e.,

Hypothesis 4: Eigen Vector Centrality will be positively correlated with adoption of scientific innovations

Users that are surrounded (are neighbors to) by more users that have adopted the tool will most likely be positively influenced to adopt the tool because of spillover effects or contagion (Leenders, 2002). Peres et al. (2010) noted that the spillover effects can be direct and indirect. The authors noted that the spillover effects will be positive if the adoption decision is directly affected by the number of immediate individual neighbors

that have adopted the tools (this could be likened to degree centrality) and indirect if the decision to adopt a tool is based on the number of indirect neighbors that have adopted the tool (this could be likened to both betweeness and closeness centrality measures). However, the difference between spillover effects to the centrality measure is the fact that interpersonal communication does not have to be present for network externalities to work (Peres et al., 2010). Autocorrelation modelling is therefore able to capture the effects of network spillovers effects on tool adoption by users and we hypothesis that the network spillover effects (multivariate dependent variables) will be positively correlated with adoption of tools, i.e.,

Hypothesis 5: Spatial Autocorrelation parameter will be positively correlated with adoption of scientific innovations.

5.3 Methodology

In this study we explored the communication channels and social structures aspects of diffusion on usage of tools (softwares) in user community of the nanoHUB.org cyberinfrastructure using macro (bass model), agent based model (discrete time hazard model) and the spatial autocorrelation model version of the discrete time hazard model. The models are specified in details below.

5.3.1 The Rate of Diffusion of Tools in the User Network: An Application of Bass Model

The bass model is an amassed model that defines the transmission of information through the behavior of the users in the network. The model is simple, tractable and incorporates social aspects into its structure (Jackson, 2008. p. 187). The Bass model explains the

mechanism of how adopters and potential adopters of a scientific innovations interact with each other in the user network (Jackson, 2008. p. 187). The model is based on the premise that adopters are innovators or imitators and the speed and timing of adoption depends on their degree of innovativeness and the degree of imitation among adopters. The bass model for continuous time period t is given by the differential equation (37),

37) $$\frac{dF(t)}{dt} = (p + qF(t))(1 - F(t))$$

Where,

$F(t)$ is the fraction of users who have adopted Tool-1

$\frac{dF(t)}{dt}$ is the rate of change of adoption of a tool or the hazard function

p is the rate of innovation

q is the rate of imitation

To solve for the unknown cumulative distribution F(T) we define $L(t)$, the conditional likelihood that a user will adopt a tool at time t by bayes formula as,

38) $$L(t) = \frac{f(t)}{1-F(t)}$$

Where,

$f(t)$ is the probability density function. Equation (38) can be written as

39) $$L(t) = p + \frac{q}{\bar{N}}(t)$$

Where,

$N(t)$ is the number of consumers who have adopted the tool by time t

$\bar{N} = \bar{N}F(t)$ is a constraint that represents the total number of users who will eventually adopt the scientific innovation; The formula for calculating $\bar{N}$ is given in equation (45) below.

Equation (38) into (39) yields (after rearrangement),

40) $$f(t) = \left[p + \frac{q}{\bar{N}}N(t)\right][1 - F(t)]$$

If we define $n(t) = \bar{N}F(t)$ as the number of users adopting a tool at time t, equation (40) can be written (after some algebraic manipulation) as,

41) $$n(t) = p\bar{N} + (q - p)N(t) - \frac{q}{\bar{N}}[N(t)]^2$$

The OLS estimates $p\bar{N}$, $(q - p)$ and $\frac{q}{\bar{N}}$ in Equation (41) can be written as a, b and c respectively. Equation (41) therefore changes to,

42) $$n(t) = a + \mathrm{b}N(t) - \mathrm{c}[N(t)]^2$$

The parameter estimates p and q were calculated were calculated from (41) and (42) as,

43) $$p = \frac{a}{\bar{N}}$$

And

44) $$q = -c\bar{N}$$

$\bar{N}$ is calculated using the quadratic equation as,

45) $$\bar{N} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Differentiating (41) with respect to t yields the predicted time it takes a tool adoption to peak

46) $$t^* = \frac{1}{(p+q)} \ln\left(\frac{q}{p}\right)$$

Solving for $F(t)$ in (37) with $p > 0$ and $F(0) = 0$ yields the cdf function,

47) $$F(t) = \frac{1-e^{-(p+q)t}}{1+\frac{q}{p}e^{-(p+q)t}}$$

Parameters p and q were calculated in weekly panels from the fitted model (42) using ordinary least squares (OLS) method. The choice of OLS over the maximum likelihood (ML) method on (42) was informed by ML shortcomings of underestimating the standard errors of estimated parameters (e.g., Schmittlein & Mahajan, 1982; Srinivasan & Mason, 1986). The authors also noted that ML only considers sampling errors and ignores all other errors. The shortcomings of estimating Bass Models with OLS method include, increased likelihood of getting biased estimates due to multi-collinearity problem caused by correlated $N(t)$ and $N^2(t)$, lack of statistical inference on estimated p, q and $\bar{N}$ because we are not able to calculate their standard errors and use of discrete time series data to estimate continuous model (bass dynamic model) which might cause time invariant bias (Schmittlein & Mahajan, 1982). Jain and Rao (1989) suggested use of any nonlinear regression method as an alternative to both OLS and ML methods but this is beyond the scope of this study that seeks to evaluate the communication channels and social structure aspects of diffusion of innovations; the structural characteristics and communication aspects will be thoroughly analyzed in the probit and spatial probit model. The estimated parameter sets (p, q and $\bar{N}$) were used to calculate and forecast

diffusion based on the cumulative function (47) (Jackson, 2008, p. 187). The 'best" cumulative function was analyzed through stochastic dominance criteria.

5.3.2 The Most and Least Adopted Tools: Stochastic Dominance Criteria

Stochastic dominance (SD) describes a set of relations that hold between two distributions (Guo, 2012). A Parameter set (A) will be first order stochastically dominated by a set (B) if $F(t)_B \geq F(t)_A$. $F(t)_A$ and $F(t)_B$are the cumulative functions (47) derived from parameter sets (A) and (B) respectively (Schmid & Trede, 1996). This implies that user networks characteristic that generate parameter set (B) lead to more diffusion of a tool than those that generate parameter set (A). Network characteristics were characterized with probit and spatial probit (autocorrelation) model estimates based on the dominating and dominated cumulative frequencies adoption curves.

5.3.3 Users and Network Characteristics that Determine Diffusion of Tools: Probit Versus Spatial Probit Models Application

We first apply a simple probit regression model that evaluates the probability that a user adopts a tool. This model is based on Banerjee et al. (2013) study that sought to evaluate the influence of opinion leaders on diffusion of microfinance in India. Banerjee et al. (2013) considered opinion leaders position in the network (communication centrality) through a logistic probability model,

48) $$y_{it} = \log\left(\frac{p_{it}}{1-p_{it}}\right) = X_i'\beta + \lambda F_{it} + v_i$$

Where,

$\log(\frac{p_{it}}{1-p_{it}})$ is the odd ratio,

X_i is the vector of covariates (representing user characteristics)

β is the vector of coefficients that describe the influence of user characteristics X_i on the odds ratio and

F_{it} is a ratio of adopting users to the total number of users that informed user i about the program. (i.e., numerator =number of users who adopts a tool and denominator =number of users who informed user i about the tool). This ratio captures the information asymmetry in the user network. If information is assumed to be perfect (where all users have the same information regarding a tool) then we can remove the ratio.

λ is the parameter representing the influence the change in the ratio of participation on the odds ratio.

v_i is user $i's$ preference shock.

Banerjee et al. (2013) noted that the preference shock v_i maybe correlated with $v_j$if say i and j are neighbors that influence each other.i.e., there might exists spatial autocorrelation in the diffusion behavior. To empirically test such a spatial auto-correlation effect, we extended Banerjee et al. (2013) model to include a spatial component but also removed the information asymmetry component because this was captured in the user network weight matrix. We also changed the depend variable from the odds ratio to a discrete binary variable. Our spatial autoregressive model therefore become,

49) $$y_{it} = \rho W y_{jt} + X_{it}\beta + \varepsilon_i$$

Where,

$y_{it} = 1$ if user i has adopted a tool at time t and 0 otherwise

$y_{jt} = 1$ if user j has adopted a tool at time t and 0 otherwise

W is an adjacent weight matrix representing relationship of the users forming the network. The edge list was constructed using an index with physical location, time of usage, start year e.t.c.

ρ is the autocorrelation parameter and all other variables are as described above but with time subscript.

Equation (48) was compared with those in equation (49) through the autocorrelation parameter ρ and correlation of preference shock v_i and v_j.

5.3.4 Data and Variables

5.3.4.1 Data

The data for this study came from the user community of the nanoHUB.Org cyberinfrastructure (Kleimeik, 2008; nanoHUB.org, 2014). Because of the enormous size of the user network, discontinuous time usage and the requirement of fully connected component, we chose users in the first half of 2006 as our sample user network. Component determine the likelihood and extent of diffusion in a network because actors (scientists) have to be linked if they are to "infect" each other (Jackson, 2008. p. 178).

Jackson (2008, p.178) noted that most studies have chosen the largest component as their network sample to study diffusion patterns because there is higher likelihood of getting infected in a more connected network structure. In this study we tried to understand the diffusion pattern of the most used tool in the class and compared it with a first order stochastically dominated tool in the user network. Bass model of diffusion was applied to evaluate the rate and structural components that determine diffusion (Mahajan et al., 1990; Jackson, 2008. p. 187). Stochastic dominance was used to determine the dominating and dominated tools by usage. The probit and spatial probit network autocorrelation models were used to identify and distinguish the user network characteristics that are highly correlated with the tool adoption in the dominating tool.

5.3.4.2 Variables

The weekly adoption time series data for the top five tools was used for the bass model. The bass model adoption curve was categorized by time to denote early versus late adopters and early versus late majority and this was used as a dependent variable for the probit and spatial probit models. The network structural characteristics were used as the explanatory variables and the central processing unit time variables from nanohub.org were used as the control variables together with the country dummy. The independent variables for the spatial probit was weight matrix of the users' connections or rate of association (digital practice activities) in addition to the network embedded characteristics that captured the local effects and control variables. All variable for probit and spatial probit model are described below.

5.3.4.2.1 The Weight Matrix.

We applied the tool user network created based probabilistic proximity index as described in Section 5.2 (Theoretical Framework and Hypothesis). This index captured users level of association largely driven by both digital practice and physical distance proximity (e.g., Matei, 2014). The largest component was used to extract the social and communication channels variables described below.

3.2.2 The network embedded variables considered included.

We calculated degree, betweeness, closeness and Eigen vector centrality measures from the largest component (described in section 5.2) as measures of as modernity, homophily, physical distance and opinion leaders. The variables definition and formulae are described in Section 3.3.2. (Variables)

3.2.3 The Control Variable

The definitions of the control variables were given from nanoHUB.org administration available at (www.nanuhub.org) and they mostly describe computing time and by extension computing capability.

Job (j.job) a job is the intensive part of tool usage, that is, the time taken to complete a given computation after parameters are set. Jobs are launched from sessions

Session Central Processing Unit time (s.cputime). This is the time spent by Central Processing Time (CPU) executing a collection of one or more processes groups (running computer programs, i.e., entering parameters, starting a job and viewing results)

Job Processing Unit walltime (j.walltime). This is the time spent by CPU executing a job (the intense part of a tool usage). Walltime is the total time taken by CPU from initiation of a program to completion.

Central Processing Unit walltime (c.walltime). Walltime is the total time taken by CPU from initiation of a program to completion. Walltime includes total time taken during that processing period.

Session viewtime (s.viewtime). This is the total time taken to access (look at) a session by users.

Job Events (j.events). These are user jobs that are being handled by nanoHUB.org API (application Programming Interphase)

Country. This is a country dummy of the location of the user (the variable has 1 if residing in US and 0 otherwise).

5.4 Results and Discussion.

We first present the macro model results (bass model) before going to the agent based models (probit and spatial probit model). Bass model results include external and internal parameter estimates, peak times, saturation levels and forecasted distributions of the 5 most adopted tools by users in first half of 2006 (Week 1-Week 26).

5.4.1 Bass Model Results

The set of complete half year data was first applied to the bass model equation (42). The cumulative number of adoptions $N(t)$ at time t and the number of adoptions $n(t)$ at time t were calculated on weekly basis as the time that a scientist started using a tool. The choice of weekly aggregation to daily or aggregation was to reproduce a graph with a

smooth and regular diffusion pattern that follows a normal and near normal distribution and which does not greatly reduce the degree of freedoms (Wright et al., 2006). OLS was used on equation (42) to solve for the external and internal influences p and q and the potential number of ultimate adopters $\bar{N}(p)$. To address the above highlighted shortcoming of using OLS technique, we first ran a correlation test between $N(t)$ and $N^2(t+1)$ in equation (42) and found no evidence of correlation in all the tools (The correlation coefficients between $N(t)$ and $N^2(t+1)$ for "pntoy", "spice3f4", "fettoy", "qclab" and "qdot" were -0.15, 0.05, 0.22, -0.09 and -0.09 respectively). Table 13 shows results of the Bass model: a, b and c are parameter estimates from OLS model (42), p is the external influence, q is the internal influence, $\bar{N}(p)$ is the potential number of ultimate adopters, q is the internal influence, $\bar{N}(a)$ is the actual number of ultimate adopters, t_1 is the period where adoptions equaled or exceeded the period which the adoption took off $(p\bar{N})$ for the first time, $t^*$is the predicted peak time and $t(a)$ the actual peak time.

Table 13: Bass Model Estimates a, b, c, p, q, $\bar{N}$ and t, and Time Series Data for $\bar{N}$ and t

Tool	a	b	c	p	q	$\bar{N}(p)$	$\bar{N}(a)$	t_1	t^*	$t(a)$
pntoy	5.71	0.13	-0.001	0.036	0.171	156.6	151	3	10.5	11
spice3f4	2.09	0.27	-0.003	0.024	0.297	87.0	84	6	13.8	14
fettoy	0.73	0.55	-0.011	0.014	0.567	51.1	58	8.5	14.8	15
qclab	0.48	0.77	-0.017	0.010	0.784	46.9	48	5	10.5	10
qdot	1.41	0.20	-0.005	0.032	0.233	43.6	43	4	11.4	11

Results show that external influence coefficient (p) is less than internal influence coefficient (q) in all tools implying that the tools are liable to adoption (Wright et al., 1997). The range of external influence was 0.01 to 0.036 for the tools while that of internal influence was 0.23 to 0.78. This range implies that external and internal

influences have varying degree of influences on each tool (e.g., Firth et al., 2006). The order of external influence on tools adoptions does not follow that of internal influence nevertheless. “pntoy” adoption has the highest external influence followed by “qdot”, “spice3f4”, “fettoy” and “qclab” in that order. “qclab” has that the highest internal influence followed by “fettoy” “spice3f4”, “qdot” and “pntoy”. The levels of internal and external influences are within the mode and range of the sum of internal and external influence coefficients, 0.5 and 0.3 to 0.7 respectively (Lawrence & Lawton, 1981).

To fit the data to the model, we applied two main methods. The first method involved calculation of the predicted peak time of adoption of tools and saturation levels and compare those with the data. Time 1 was set as the period where adoptions equaled or exceeded the period which the adoption took off ($p\bar{N}$) for the first time (Bass, 1969; Firth et al., 2006; Wright et al., 1997). The predicted and actual peak time of adoption and total number of adopters (saturation levels) are shown in Table 1 as t* and t (a) and $\bar{N}(p)$ and $\bar{N}(a)$ respectively. In the second method, we calculated the predicted (forecasted) adoption rates of tools using the external (p), internal (q) and the potential number of ultimate adopters ($\bar{N}$) estimates from (42) and (45) and compared that with the actual adoption rates through visual (graphs) and non-parametric test statistics (Kolmogorov-Smirnov (KS test)). Other studies have applied basic parametric test statistics, t-test based on an assumption of normal distribution and/ or sum of square difference between the two distributions with a lower one implying a better model fit (e.g., Firth et al., 2006). KS test statistics was used to evaluate the overall goodness of fit of the predicted versus actual distributions because our data did not follow a normal distribution. The adoption curve of the 5 top adopted plots is presented in Figure 12

below. The plots of tool usage adoption rates and cumulative rates for the 5 top adopted tools in the first half year are presented in Figures 13-17 below.

Results in Table 13 show that the actual and predicted time peaks and saturation levels are very close to each other for all the tools implying that our model fits the data pretty well (e.g., Firth et al., 2006; Wright et al., 1997). The peak time and saturations levels fits the data very well but shows some variations based on the tools.

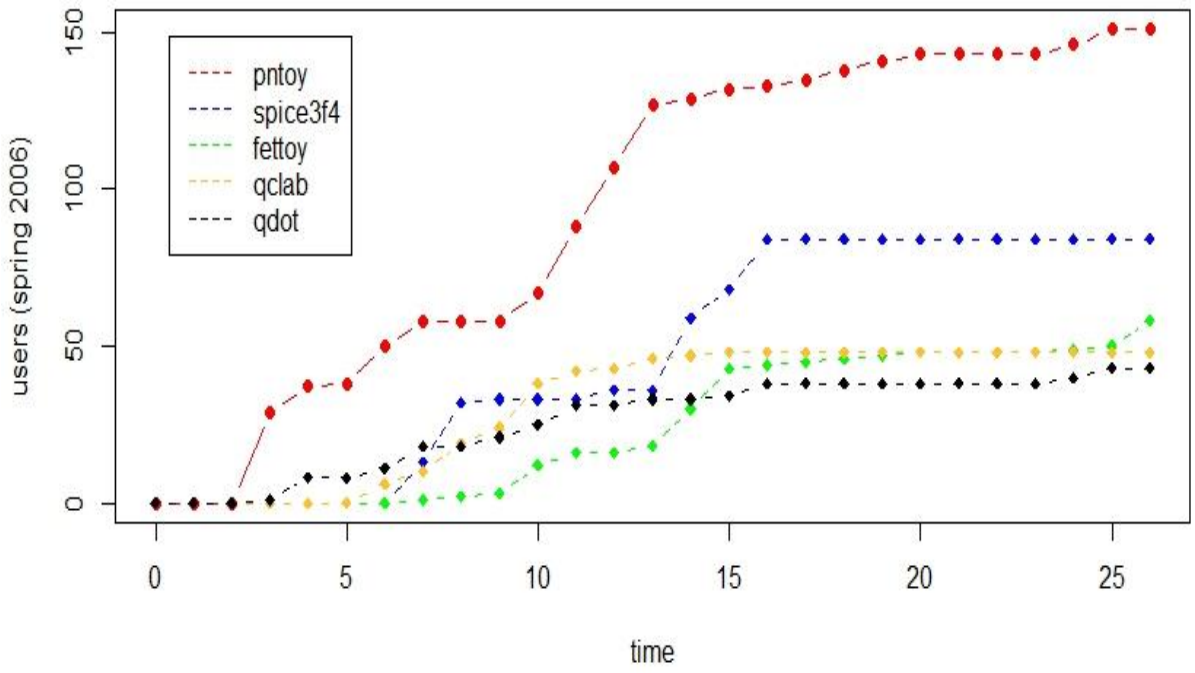


Figure 12: Adoption Curves of the 5 Most Used Tools in First Half of 2006

The adoption curves in Figure 12 seem to follow distribution that resembles logistic curve albeit with varying degree of curvature. Adoption of "pntoy" is characterized by early take off and peaking times while "fettoy" has a late take off but peaks very quickly.

“qdot” has the lowest peak off and saturation levels. “spice3f4” and “qclab” seem to pick later than “pntoy” but are above “qdot”. To further understand the relationship between these distributions we ran a stochastic dominance test and results are discussed in the following section. The weekly frequency and cumulative distributions of individual tools bass model adoption and time series data are presented in Figures 13-17 below.

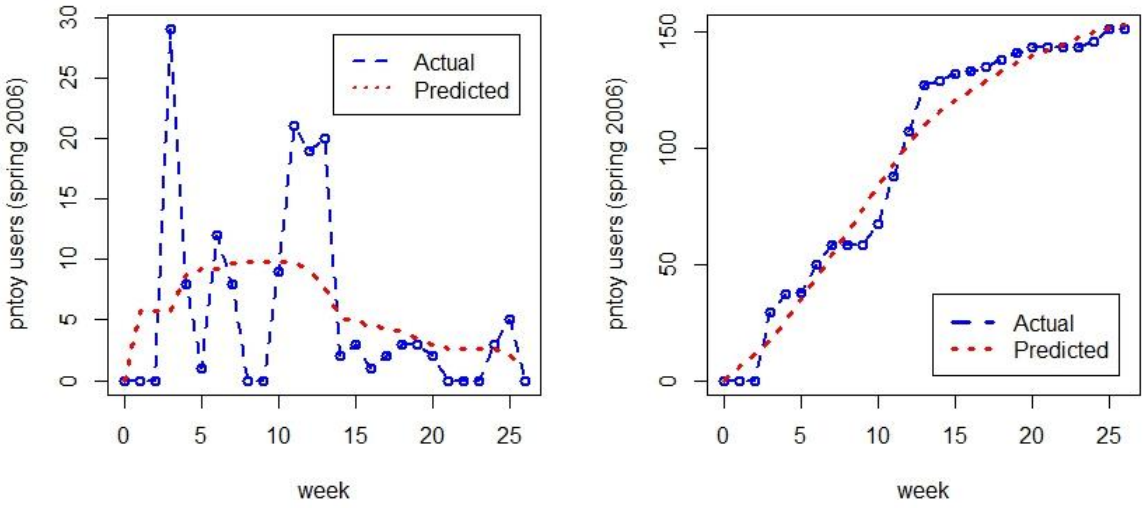


Figure 13: Fitting Bass Model Estimates to Data (Time Series) for "pntoy" Tool for 1st half of 2006

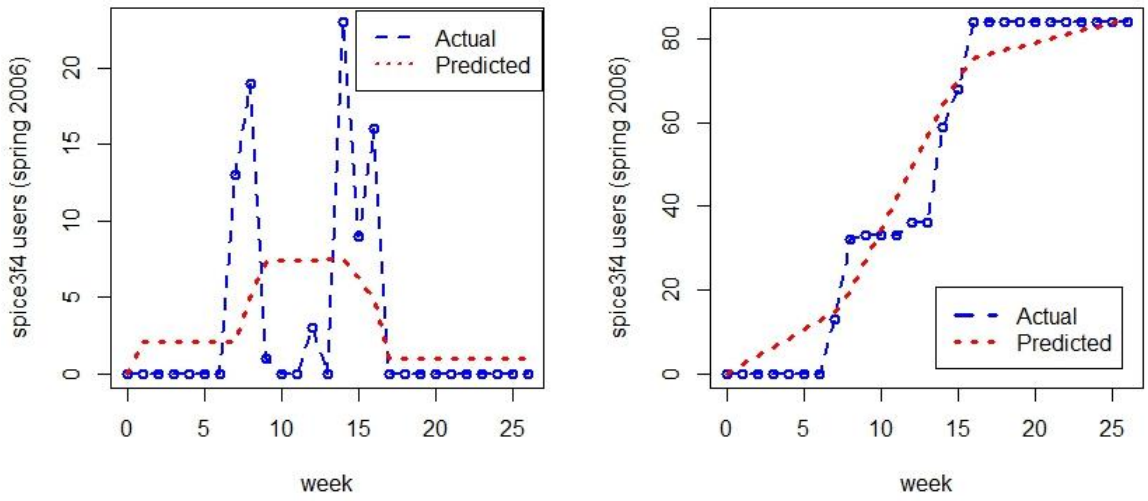


Figure 14: Fitting Bass Model Estimates to Data (Time Series) for "spice3f4" Tool for 1st half of 2006

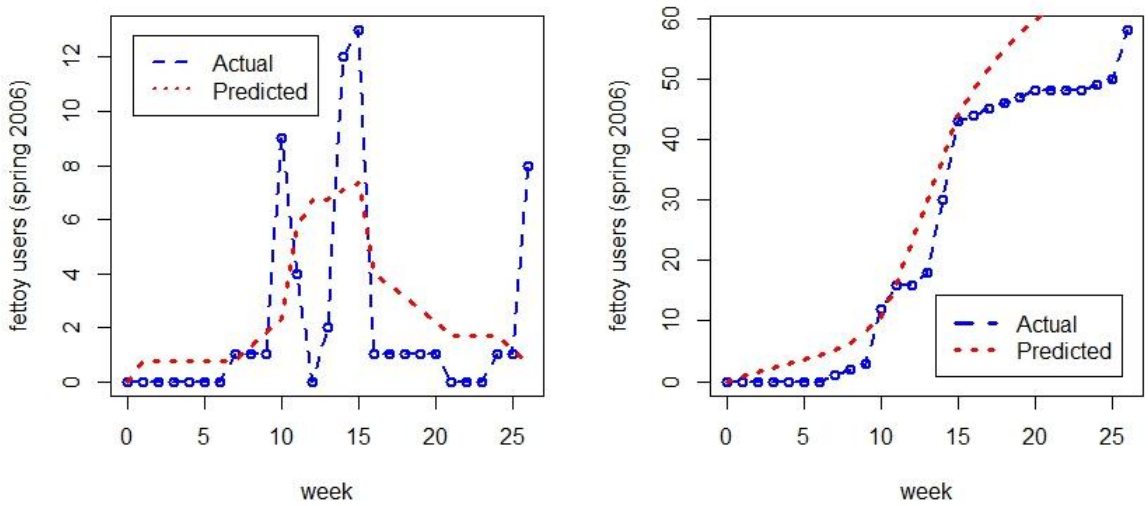


Figure 15: Fitting Bass Model Estimates to Data (Time Series) for "fettoy" Tool for 1st half of 2006

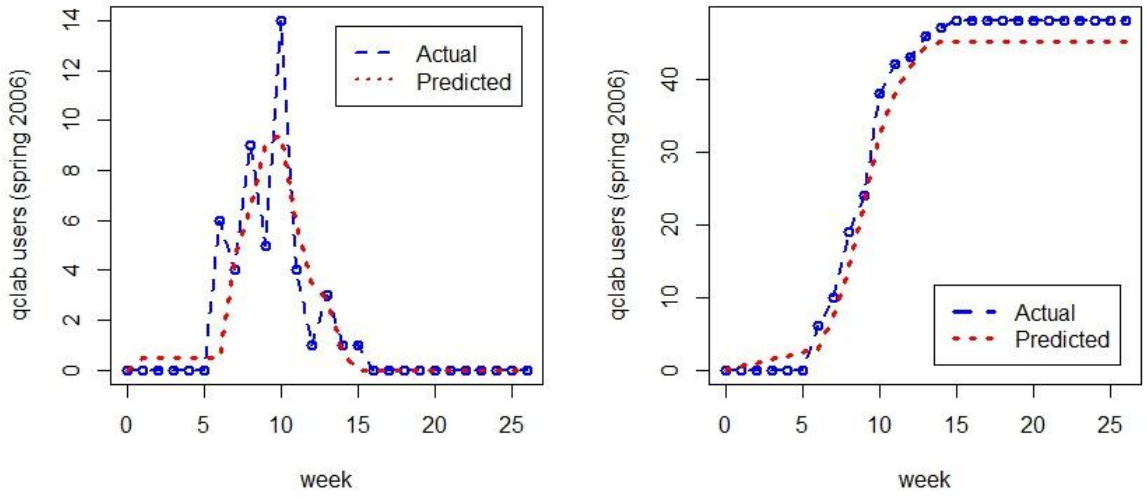


Figure 16: Fitting Bass Model Estimates to Data (Time Series) "qclab" Tool for 1st half of 2006

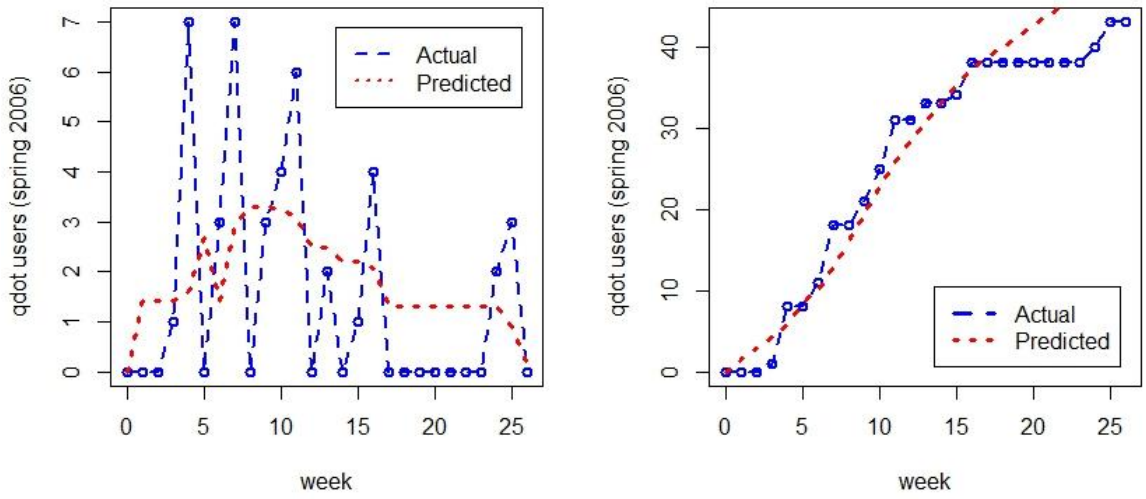


Figure 17: Fitting Bass Model Estimates to Data (Time Series) for "qdot" Tool for 1st half of 2006

Figures 13-17 shows relatively fitting but varying bass model to data. The fit is better seen in bass model and time series cumulative frequency distributions that has less noise than weekly frequency distributions. KS test statistics was further used to evaluate the overall fit of the distributions the bass model distributions and the time series data. KS test has the null hypothesis that the bass model distribution followed the time series data distribution. Results are shown in Table 14 below.

Table 14: KS Test for Goodness of Fit for Bass Model and Time Series Data Distributions

Tool	**Difference (Obs-Pred) in Distributions**	**p-value**
Pntoy	0.148	0.928
Spice3f4	0.37**	0.049
fettoy	0.333	0.100
qclab	0.519***	0.001
qdot	0.259	0.324

*,** denotes 1% and 5% significance levels

Results of the KS test in Table 14 shows the difference in distributions of the bass model and the time series data and the p-values. Results show that “spice3f4” and “qclab” has a statistically significance difference in distributions and we fail to reject the hypothesis that the model and data distributions followed each other. Other tools show that the model distribution followed data distribution.

5.4.1.1 Pairwise Stochastic Dominance Test for Adoption Curves Distributions

KS-test was also used to evaluate pairwise difference of tool adoption curves. Table 15 shows the KS-test results for stochastic dominance. The first column shows the distribution that is to be evaluated (treatment distribution) against the reference

distribution (control distribution) in the second column. Results of pairwise stochastics dominance of the distributions are presented in Table 15 below

Table 15: Pairwise Stochastic Dominance of the Distributions

Distr. 1	Distr. 2	Difference (1-2) in Distributions	p-value
Pntoy	Spice3f4	0.593***	0.000
	fettoy	0.704***	0.000
	qclab	0.778***	0.000
	qdot	0.778***	0.000
Spice3f4	fettoy	0.482**	0.004
	qclab	0.482**	0.004
	qdot	0.482**	0.004
Fettoy	qclab	0.222	0.517
	qdot	0.407**	0.023
Qclab	qdot	0.519***	0.046

Results show that distribution of “pntoy” stochastically dominates all other tool distributions. Results also show the distribution of “spice3f4” stochastically dominates all other tools but “pntoy”. “Fettoy” and “qclab” stochastically dominates “qdot”. The highest difference in dominated distributions is found between “pntoy” and “qclab” and “qdot” but “spice3f4” shows consistent difference in distributions to “qclab” and “qdot”. We thus choose a stochastically dominating distribution “spice3f4” and dominated “qdot” and map out the distinguishing network characteristics that might be responsible for the tool adoption in the “spice3f4” (the distributions of the model to data had also very good fit of the model. See figures 14 and 17).

Probit and Spatial Probit models estimates were used to map out the distinguishing network characteristics based on the above highlighted results of stochastic

dominance between “spice3f4” and “dqot”. The model of the dominated distribution “qdot” was taken as the control while the dominating distribution “spice3f4” was taken as the treatment. By the stochastic dominance we will be able to map out the distinguishing network characteristics of users in those distribution which will give the direction of the user network characteristics responsible for tools uptake including the degree, closeness, betweeness and eigen vector centrality. Results of the network characteristics of the two distributions are shown in Table 16 below.

Probit and Spatial Probit models were based on the assumption that all users in the nanoHUB.org were aware (or had access to information) of the tools available for use and that the choice of adoption was purely based on individual scientist’s preferences that would be partly determined by the level of digital practice. However, because of communication channels and social structure influence we anticipate correlation in the error term estimates of the logistic function (e.g., Banerjee et al., 2013). The correlated error terms will lead to biased estimates and we extended the probit model to capture the network autocorrelation effect through and Spatial probit model that adds a weighted neighborhood matrix.

5.4.2 Results of the Probit and Spatial Probit Model

We compare the distinguishing network and communication channels characteristics responsible for adoption in the stochastically dominating “spice3f4” and dominated “qdot” for early adopters and early majority adopters. The early adopters and early majority users are based on Rogers (1983) classical 5 phase categories. Our data does not follow a logistic or near normal distribution but we categorize early adopters as 16% of the tools users by time t and early majority as 50% of the users by time t and evaluate the

distinguishing network characteristics (including communication channels) and personal characteristics influencing the diffusion process (Rogers, 1983). We do not consider the innovators (2.5%) because of the small size of our component data set. However, by considering the communication channels and network characteristics responsible for the tool adoption, we will be indirectly testing the external and internal influence whereby absence of significance estimates implies lack of the said social influences. Tables 16, 17 and 18 show the descriptive statistics of the model variables and results of probit and spatial probit regressions models.

Results show that 12% percent of the total users had adopted “spice3f4” tool while about 1% had adopted “dqot” by week 4 (16% of the term lifecycle). Results also show that about 12% and 3% of the total users had adopted the “spice3f4” and “pntoy” at week 13 (50% of the term life cycle). Table 16 also shows the user network structural characteristics that were used as independent variables for our models. Results show the mean of Bonacich centrality to be -0.75 and the standard deviation to be low 0.046. Results imply that the network is characterized by less powerful or influential users/leaders because Bonacich centrality measures the number of influential people/leaders in the network (e.,g Jackson, 2008). A similar and related centrality measure of leadership/influential persons in the network is Eigen vector centrality and google page rank. Eigen vector centrality had a low mean of 0.36 and standard deviation 0.033 while google page rank had a mean of 4.79 with a standard deviation of 0.08. The results seem to confirm that the user network is characterized by less influential persons/ leaders/decision makers.

Table 16: Descriptive Statistics of Probit and Spatial Probit Model Variables

early (e') Adopters	n	Mean	Std. Err	Min	Mid	Max
pntoy-e	209	0.115	0.022	0	0	1
spice3f4_e	209	0.120	0.023	0	0	1
fettoy_e	209	0.005	0.005	0	0	1
qclab_e	209	0.038	0.013	0	0	1
qdot_e	209	0.005	0.005	0	0	1
Early Majority (em) Adopters						
pntoy_em	209	0.120	0.023	0	0	1
spice3f4_em	209	0.124	0.023	0	0	1
fettoy_em	209	0.010	0.007	0	0	1
qclab_em	209	0.105	0.021	0	0	1
qdot_em	209	0.029	0.012	0	0	1
Independent Variables						
Bonacich Centrality	209	-0.751	0.046	-3.08	-0.76	1.55
Betweeness Centrality	209	1.912	0.155	0	2.48	9.28
Closeness Centrality	209	0.815	0.009	0.46	0.87	1.15
Degree Centrality	209	86.105	4.011	2.00	82.00	184.00
Eigen Vector Centrality	209	0.360	0.033	0.00	0.00	1.00
Google Page Rank	209	4.785	0.081	0.93	4.87	9.15
Control Variables						
s.cputime	209	2.149	0.098	0.0	2.24	4.62
j.cputime	209	1.182	0.087	0.0	0.64	4.62
c.walltime	209	7.203	0.177	0.0	7.33	15.55
s.viewtime	209	6.258	0.230	0.0	7.07	15.55
j. job	209	0.541	0.035	0	1	1
j.event	209	0.459	0.035	0	0	1
Country	209	0.842	0.025	0	1	1

Results also show the mean betweeness and closeness centrality measures to be 1.91 and 0.815 while the standard deviations to be 0.155 and 0.009 respectively. Betweeness centrality measures the average span across the network structural holes and it is a

measure of easiness of information passing to the peripheral users in the network (Valente et al., 2008). Betweeness centrality measure indicates that there is both direct and direct information passing in the network. Results imply that there is relatively high levels of direct and indirect information passing in the network. Closeness centrality measures the average reachability (closeness) between users in the network and it is an indication of network efficiency and independence in transmitting information. i.e., users transmit information efficiently because of close proximity and are therefore independent because they do not reach out to peripheral users for information (Freidkin, 1991; Valente et al., 2008). The low closeness centrality measure imply that the network is less efficient in transmitting information and users are therefore dependent on other users in getting information. The user network has a mean degree centrality of 86 and a low standard deviation, 4.0. This implies that users in the largest component have a relatively high degree of connectedness and we expect high information exchange. Valente et al. (2008) noted that degree centrality is highly correlated with closeness centrality because the two measures are directly linked to direct and efficient information exchange.

The control variable used in the analysis included the tool usage time variables comprising the time spent running or viewing an application or simulating a program, job, processes or session. These are measured from the central processing units and or the user interphase view time and are measures of the nanoHUB.org computing power by users. Results show relatively short use time with some variability that was of interest for statistical analysis. For example, the average time taken to run a session was 2.149 seconds with a low standard deviation of 0.098 while the time taken a job is about half of the time 1.18 seconds. The average time central processing unit walltime (total time to

run a program from initiation to completion) was 7.2 seconds with a standard deviation of 0.177 while the average time taken to view a session by users was 6.26 seconds with a standard deviation of 0.23. The other control variable was the geo-location of users, country dummy. The country dummy shows that about 84% of the users are located in the US. Other geolocation variables that had some variations were not considered because they were included in constructing the actual weight matrix.

5.4.2.1 Probit and Spatial Probit Models

Results of probit and spatial probit models for the early adopters and early majority users for "spice3f4" and "qdot" tools are presented in Tables 17 and 18 respectively. The dependent variable in Tables 17 and 18 is a binary variable (with 1 and 0) representing the number of tool adopters at time corresponding 16% and 50% of the distribution respectively. The variable has 1 if the user had adopted the tool at that particular time and 0 otherwise. Explanatory variables included the network characteristics representing the communication channels and social structure characteristics and control variables representing individual tool usage time and country variables. We first evaluated the model fit using likelihood ratio (LR) test and also tested the residuals for autocorrelation using Moran's I test. The LR test had the null hypothesis that the log likelihood of the restricted and unrestricted models is not different from zero. The LR test results show that all 8 model results fit the data because the log likelihood of restricted and unrestricted models are different from zero. Moran test for residual autocorrelation on probit models confirms that the error terms are autocorrelated implying that social influence is present in diffusion of tools.

Table 17: Probit and Spatial Probit Results of Dominating "spice3f4" Tool and Dominated Tool "qdot" Early Adopters

	spice3f4_e		qdot_e	
Variable	Probit	S.Probit	Probit	S.Probit
	0.130	-4.890**	-0.022	-9.228***
intercept	(0.158)	(2.297)	(0.037)	(2.045)
	-0.114**	-1.753***	0.003	0.606
Bonacich Centrality	(0.036)	(0.462)	(0.008)	(0.492)
	0.039**	0.287**	-0.004	-0.229
Betweeness Centrality	(0.012)	(0.143)	(0.003)	(0.143)
	-0.154	-0.784	0.028	2.104
Closeness Centrality	(0.198)	(2.115)	(0.047)	(2.114)
	0.002***	0.025***	0.000	-0.004
Degree Centrality	(0.000)	(0.007)	(0.000)	(0.009)
	-0.086**	-1.042**	0.007	0.395
Google Page Rank	(0.028)	(0.341)	(0.006)	(0.314)
	0.026	0.202	0.002	0.145
s.cputime	(0.026)	(0.181)	(0.006)	(0.290)
	-0.035	-0.335	0.001	-0.047
j.cputime	(0.023)	(0.209)	(0.005)	(0.231)
	0.012	0.139	0.000	-0.055
s.walltime	(0.011)	(0.090)	(0.003)	(0.116)
	0.117**	1.022*	0.006	-0.183
j.event	(0.059)	(0.598)	(0.014)	(0.810)
	0.040	1.773	-0.034**	-0.444
Country (US==1)	(0.064)	(1.223)	(0.015)	(0.513)
		-0.010**		-0.050***
rho		(0.003)		(0.011)
Morans I residual test	3.42***		7.41***	
loglik		-44.58		-16.05
AIC	94.67	113.16	-510.01	56.1
LR		81.92***		116.39***

The presence of social influence is confirmed by the spatial autocorrelation parameter of the spatial probit models. Spatial probit models corrects the autocorrelation bias through

inclusion of a weighted neighborhood influence variable (Lesage & Pace, 2008). The spatial autocorrelation parameter estimates for “spice3f4” and “qdot” for early adopters was -0.010 and -0.050 at 5% and 1% significance levels while it was -0.008 and -0.010 at 5% significance levels for early majority. These results imply that user network has a negative spillover effect of diffusion of tools in the largest component, that is, being embedded in the largest component reduced the probability of adopting a tool. The negative spillover effect in the largest component could be attributed to the communication channels and structure of the network and the above highlighted network structural characteristics. The network structural characteristics that enable/facilitate communication are further discussed for the early adopters and early majority below. As aforementioned, we largely expect communication channels and social influence to be absent amongst early adopters than late adopters because there is a low probability of adoption from social influence given the small number of tool adopters at 16%.

Results show that most of the communications and network structural characteristics that facilitate communication are significant in outlining increased probability of tool adoption for the dominating distribution “spice3f4” to dominated “qdot” for early adopters of tools but not for early majority users. Table 17 shows that having high degree and betweeness centrality increases the probability of adopting a tool while high Eigen vector and bonacich centrality decreases the probability of adopting a tool in the dominating “spice3f4” and not the dominated “qdot”. These results imply that the probability of adopting a tool is increased by network internal factors (network characteristics) and not just external factors. Our results are supported by Borgatti and Halgin (2011) Banerjee et al. (2013) analytical and empirical papers that “found” that

social influence or contagion is a factor that lead to diffusion of innovation. Users that have high number of connections (high degree centrality) have a higher chance of adopting a tool because degree centrality facilitates direct transmission of influence or information that might lead to the adoption decision (Valente et al., 2008). The author notes that this is a measure of network efficiency and independence because users take a relatively "shorter" time to transmit information and do not need third parties to get that information. A related centrality measure of network efficiency is closeness centrality but results showed this measure to be insignificant in this study. Results also show that users that have high betweeness centrality (users that lie between paths of others-brokers) have a higher probability of adopting a tool. These results imply that such users are able to get relevant information about a tool from direct and indirect sources by the virtue of their position and this might influence their decision in adopting a tool (Valente et al., 2008). While we would expect users that are connected to leaders and/well connected users to have a higher probability of adopting a tool because of enabled/facilitated linkages to other users, results show that this actually decreases the probability of adoption of tools. This is confirmed by the negative and statistically significant Eigen, bonacich and googlepage rank parameter estimates that measure effect of influence/ power. Tables 16 also shows only j.event control variables has an effect of increasing the probability of tool adoption for dominating "spice3f4" adoption amongst the early adopters and early majority users but for all models but spatial probit model of "spice3f4". Results imply there is a higher likelihood of tool adoption by users that run their applications or jobs when there are many jobs running simultaneously. J.events measures the number of jobs running on the nanoHUB.org API. J.event measure the jobs that are being handled by the

nanoHUB.org API (application Programming Interphase). Table 18 also shows that the time spent by Central Processing Time (CPU) in executing a job decreases the probability of adopting a tool. i.e., the more time is spent running a job will lead to reduced likelihood of adoption of a tool.

Table 18: Probit and Spatial Probit Results of Dominating "spice3f4" Tool and Dominated "qdot" Tool Early Majority Users

	spice3f4_em		qdot_em	
Variable	Probit	Spatial Probit	Probit	Spatial Probit
intercept	0.133 (0.160)	-4.258** (2.076)	0.130 (0.158)	-4.890** (2.297)
Bonacich Centrality	-0.114** (0.036)	-2.03*** (0.374)	-0.114** (0.036)	-1.753*** (0.462)
Betweeness Centrality	0.040*** (0.012)	0.306 (0.203)	0.039** (0.012)	0.287** (0.143)
Closeness Centrality	-0.135 (0.201)	-1.349 (2.502)	-0.154 (0.198)	-0.784 (2.115)
Degree Centrality	0.002*** (0.001)	0.030*** (0.006)	0.002*** (0.000)	0.025*** (0.007)
Google Page Rank	-0.090** (0.028)	-1.25*** (0.316)	-0.086** (0.028)	-1.042** (0.341)
s.cputime	0.029 (0.026)	0.157 (0.170)	0.026 (0.026)	0.202 (0.181)
j.cputime	-0.036 (0.024)	-0.386* (0.215)	-0.035 (0.023)	-0.335 (0.209)
s.walltime	0.011 (0.011)	0.107 (0.092)	0.012 (0.011)	0.139 (0.090)
j.event	0.113* (0.060)	0.858 (0.525)	0.117** (0.059)	1.022* (0.598)
Country (US==1)	0.039 (0.065)	2.321** (1.073)	0.040 (0.064)	1.773 (1.223)
rho		-0.008** (0.003)		-0.010** (0.003)
Morans I residual test	3.48***		3.42***	
loglik		-46.07		-44.58
AIC	100.21	116.13	94.67	113.16
LR		60.28***		121.67***

Other significant control variable was country data for the dominated “qdot” probit model amongst the early majority tool adopters. The country dummy shows that being a non us citizen increases the probability of adopting a tool of for the qdot tool.

5.5 Summary and Concluding Remarks

This study sought to understand diffusion of tools amongst scientific users in an online community. Diffusion of innovation theory was explored from both a macro and micro modelling perspective. The macro model was used to understand and rank usage of tool amongst users in an aggregate manner because users are assumed to be similar in their adoption preferences (homogeneous). The bass model determined the external, internal factors influencing adoption of tools and also forecasted adoption in online community based on estimated parameters. Micro models were used to complement Bass model and also understand the actual network structural and hence communication channels that were responsible for adoption of tools which showed different adoption patterns. The aggregate assumption of the global social influence in bass model was further tested using an autocorrelation model. As such probit and spatial probit models were used as the micro economics models.

Data came from user network of nanHUB.org cyberinfrastructure that brings together user community across the globe through online high speed internet and high capacity computers. The time series rate of adoption was used as the data for the Bass model. Data for micro models included a binary rate of adoption as the dependent variable, a weight matrix (adjacency matrix) that was constructed based on close proximity to evaluate the social contagion influence aspects of the network and the Network structural characteristics as explanatory variables and some usage variables as

control variables. Results show that bass model is a good predictor for tool adoption in an online community setting. Results also show different tools to have varying tool usage rates, external and internal influences, time of peak and saturation levels. Both external and internal factors were found to be responsible for tools adoption. Results of the micro-based model found degrees and betweeness centrality as some of the internal variables that influenced the adoption process positively while centrality measures of power or leadership were found to have negative influence of adoption. The job usage time was also found to negatively influence diffusion.

While these results seem inconclusive, for a start, we have seen that diffusion process in online communities also exhibit patterns similar to market based innovation which is the main theoretical contribution. In particular, bass model was found to fit and thus predict the diffusion process pretty well. While we might not come up with a particular value for external and internal influence, results fell in the range found in market goods and this is an important practical contribution that is useful to platform managers. Therefore, we can recommend policy to apply bass model to forecast adoption and also determine the probable timing and saturation levels of tools in an online setting based on the standard 0.5 mode value of external and internal influences but allow some variations. Forecasted values can be good for determining the required CPU capacity and the possible peak time can be used to determine when and when not to put more awareness effort, say of advertisements and flyers. Another theoretical contribution was the finding of degree and betweeness centrality as having a positive influence on probability of tool adoption but not leadership in the micro models. This finding has also practical contribution to platform managers whom we would recommend to

enable/enhance activities that will encourage more direct connection and communications, like live chats and also enable forums for reaching out to other others users in an online questions and answers setting on the basis of encouraging both direct and indirect connections (i.e., increasing betweeness centrality). Another practical contribution was the revelation that the time of running a job discouraged adoption of tools. This implies that the platform managers (administrators) needs to works on ways of reducing the time of running a job. For example, the administration can try cloud computing or increase the CPU capacity to increase the speed of running a job. The projection of the capacity and cloud computing can very much be determined by the predictions of the bass model.

CHAPTER 6. CONCLUSIONS

Scientific collaborations have witnessed major changes in the last two decades because of progression in technological communication (mostly high speed internet). The changes have transformed systems of digital practice including the "traditional" research and collaboration methods in various field of science where collaboraties have increased in size and frequency. Online collaboraties are now larger and operate in a more efficient manner that is believed to increase productivity; innovations and self-growth for participants (Brunswicker et al., 2015; Gonzalez-Brambila et al., 2013; Matei, 2014; Schroeder et al., 2007). Research examining this new phenomenon have focused on understanding mechanisms of online collaboraties that influence output and how the networks collaboraties form. This dissertation is focused on understanding the formations and effect of such kind of online communities (using the nanoHUB.org cyberinfrastructure) to members.

6.1 Summary of Papers

Network theory is used to determine the effect of networks on members' productivity while theory of network is used to understand how the online communities are forming. Several sub-theories of network theory were considered in understanding these phenomena. These include, social exchange, small world, structural holes and

strength of weak ties, theory of collective action, random and preferential attachment and diffusion theories.

The first paper applied network theory and spatial econometrics technique to evaluate how scientist's positioning in digital spaces correlated with his/her productivity. The second paper looked at the network formation mechanism using theory of network. The third paper, like the first, applied network theory and spatial econometrics to understand user network characteristics that influence diffusion of scientific tools in the user network.

6.1.1 Conclusion for Paper One: Embeddedness in Multiple Network Spaces on Scientist Development; Higher Order Spatial Models and Network Fixed Effect Models

This study evaluated network local and global structural and relational factors that influence participating member's digital practice capital and hence productivity in a developer community. The global spatial autocorrelation parameter was found to be negative and statistically different from zero implying that there is a negative spatial spillover effect on digital practice capital in the developer network. The negative spillover effects was attributed to model representation and the characteristics of the chosen weight matrix/matrices. Both weight matrices are characterized by high clustering[21] (small worlds) but do not show homophily amongst those clusters (low

[21] Clustering coefficient for both weight matrices was 0.76 while assortativity (homophily) measure for weight matrices 1 and 2 was calculated as -0.0075 and -0.0026 respectively (See Table 4 in section 4.2)

assortativity coefficient). The practical implication of these results is the revelation that developers in both developer and authorship network cluster not based on similar developers but other factors that could be work related. The weight matrices were also characterized by low density and relatively low reciprocity. High clustering, low density and reciprocity encourages developers to span structural hole while searching for non-redundant knowledge from "trusted" (reliable) developers that will give them leverage to acquire digital practice capital to develop quality tools that have a high probability of getting a cite (e.g., Burt 1992; 2004; Hargadon & Sutton, 1997).

The local structural network characteristics of eigen vector centrality had statistically significant effects on probability of getting citations. Eigen vector centrality measures the developer's position relative to influential/highly accomplished developers in the network. Results showed that being close to influential developers in the network increases the digital practice capital and hence the probability of getting a citation. This finding supports the emerging new school of thought which argues that the "type" of scientist that a developer associates/works with might influence citation of developed tools (Gonzalez-Bambrila, 2013). This finding is a major theoretical contribution that supports the emerging new school of thought which argues that the "type" of developers that a developer associates/works with might influence citation of developed tools (Gonzalez-Bambrila et al., 2013).

Results also showed that developers that are in more than one network spaces had a higher probability of being successful than those that were in one. These results are also a major practical contribution in digital practice organization since they reveal that being

embedded in multiple networks increases the chances of developing a tool that will get citation.

6.1.2 Conclusion for Paper Two: Growing Developer Community in Scientific Digital Ecosystems: Exponential Random Graph Models

This study evaluated the network formation, operation and organization (collaboration) and sustenance mechanism in an online enabled cyberinfrastructure (nanoHUB.org) through social network modelling. A simple link to link network formation model was used to evaluate the network formation pattern. Stochastic dominance model was used to evaluate the most efficient model which was used to evaluate and fit the network characteristics that are important for developer networks. ERG (P*) model was used to compare and validate the network formation characteristics of the developer network. The study was anchored in theory of network that mostly explains the patterns of the network formation. Other network self-organizing and sustenance sub-theories including tendency for the networks to show reciprocity and clustering were also tested in the model. Both link to link and ERGM models results show that developers form in a manner that follow a pure uniform random distribution. The practical implication of this study is that online platform managers should put least efforts in activities that try to influence membership to communities. The theoretical implication of this results is the revelation that a simple link to link model performs just as good as any other ERGM in determining the patterns of formation and organization of networks. Other theoretical implication is the characterization of network characteristics from the most efficient degree distribution that is derived from stochastic dominance criteria. Results also show that developers are characterized by low tendencies to

reciprocate but have a high tendency to form clusters. These results imply that developer's participation in online communities is not exhibited by back and forth exchanges of coding but flows exchanges that coalesce (cluster) in small groups. These results imply that platform managers should not engage in activities that might enhance to direct exchanges through the SVN files. Results also show that developers show low tendencies towards homophily, that is, developer network exhibits heterogeneous coders working on a particular tool.

6.1.3 Conclusion for Paper Three; Communication Channels and Social Structures Aspects of Diffusion of Software in Online Digital User Community: Bass Model and Network Autocorrelative Micro Modelling

This study sought to further understand the communication channels and social structures aspects of diffusion of tools amongst scientific users in an online community. Results show that bass model is a good predictor for tool adoption in an online community setting. Results also show different tools to have varying tool usage rates, external and internal influences, time of peak and saturation levels. Both external and internal factors were found to be responsible for tools adoption. Results of the micro-based model found degrees and betweeness centrality as some of the internal variables that influenced the adoption process positively while centrality measures of power or leadership was found to have negative influence of adoption. The job usage time was also found to have negative significance on diffusion.

While these results seem inconclusive, for a start, we have seen that diffusion process in online communities also exhibit patterns similar to market based innovation.

Bass model was found to fit and thus predict the diffusion process pretty well. While we might not come up with a particular value for external and internal influence we can certainly say they range fall within the range found in market goods. Therefore, we can recommend policy to apply bass model to forecast adoption and also determine the probable timing and saturation levels of tools in an online setting based on the standard 0.5 mode value of external and internal influences but allow some variations. Forecasted values can be good for determining the required CPU capacity and the possible peak time can be used to determine when and when not to put more awareness effort, say of advertisements and flyers.

The micro models have found that degree and betweeness centrality as having a positive influence on increased probability of adoption but not leadership. We therefore recommend policy to enable activities that will encourage more direct connection and communications, like live chats and also enable forums for reaching out to other others users in an online questions and answers setting on the basis of encouraging both direct and indirect connections (i.e., increasing betweeness centrality). Results also found that the time of running a job discourages adoption of tools. This implies that the administration needs to works on ways of reducing the time of running a job. For example, the administration can try cloud computing or increase the CPU capacity to increase the speed of running a job. The projection of the capacity and cloud computing can very much be determined by the predictions of the bass model.

6.2 Limitations and Future Work

Future Work could improve all the papers by considering various alternatives of scientist association captured by the weight matrix. In the first paper, a simple gravity model based on the level of code modification deletion and time of association was used to generate the weight matrix. This formula generated a weight matrix that was fully connected and very dense. The weight matrix formulation and characteristics were attributed to giving negative spillover effects. Different alternatives methods of generating the association of scientist could be devised to improve the weight matrix. Rewiring technique could also be considered as an alternative and result compared to get a matrix that is ideal and representative of the real world. Our scientist data also did not have several personal covariates that would greatly improve the study finding. Further studies with these covariates could be applied to control for the actual effect of the network with those of individual scientist.

The second essay considered the network formation, operation and sustenance mechanism for a period of 8 years using a simple link to link model and exponential graph modelling. The weight matrix was also constructed using scientist association based on the level of work they put on the codes using the gravity model. The weight matrix could also be improved through different formulation of association and or through rewiring. The study could also explore the game theory aspect of network formation to better understand the actual components driving the network formation process.

The third essay considered diffusion of tools in one term because of discontinuity in the terms and tools. The study did not also consider the tool injection point while

evaluation the diffusion process because of lack of that data. Future work could consider the injection point of tools because it is one of the main drivers of diffusion of innovation. The study could also be improved through addition of individual user's characteristics because these will largely help to control and distinguish the main drivers of diffusion of tools in digital platforms. The diffusion patterns and trends of other terms could also be considered to generate a more precise range of external and internal influence.

BIBLIOGRAPHY

BIBLIOGRAPHY

Abbasi, A., Chung, K. S. K., & Hossain, L. (2012). Egocentric analysis of co-authorship network structure, position and performance. *Information Processing & Management, 48*(4), 671-679. doi:10.1016/j.ipm.2011.09.001

Abbasi, A., Hossain, L., & Leydesdorff, L. (2012). Betweenness centrality as a driver of preferential attachment in the evolution of research collaboration networks. *Journal of Informetrics, 6*(3), 403-412. doi:10.1016/j.joi.2012.01.002

Abelson, P. H. (1980). Scientific Communication. *Science, 209*(4452, Centennial Issue (Jul. 4, 1980), pp. 60-62), 60-62. Retrieved from http://www.jstor.org/stable/1684837

Albert, R., Hawoong, J., & Albert-László, B. (1999). Diameter of the World-Wide Web. *Nature, 401*, 130. Retrieved from www.nature.com

Anderson, J. E. (2010). THE GRAVITY MODEL. *NBER WORKING PAPER SERIES*. Retrieved from http://www.nber.org/papers/w16576

Anselin, L. (1988). *Spatial Econometrics: Methods and Models*: Springer.

Badinger, H., & Egger, P. (2011). Estimation of higher-order spatial autoregressive cross-section models with heteroscedastic disturbances. *Papers in Regional Science, 90*(1), 213-235. doi:10.1111/j.1435-5957.2010.00323.x

Ballester, C., Antoni, C.-A., & Yves, Z. (2006). Who's Who in Networks. Wanted: The Key Player. *Econometrica, 74*(5), 1403-1417. Retrieved from http://www.jstor.org/stable/3805930

Banerjee, A., Chandrasekhar, A. G., Duflo, E., & Jackson, M. O. (2013). The diffusion of microfinance. *Science, 341*(6144), 1236498. doi:10.1126/science.1236498

Barabási, A.-L., & Albert, R. (1999). Emergence of Scaling in Random Networks. *Science, 286*(5439), 509-512. Retrieved from http://www.jstor.org/stable/2899318?seq=1&cid=pdf-reference#references_tab_contents

Barzilai-Nahon, K. (2009). Gatekeeping: A critical review. *Annual Review of Information Science and Technology, 43*, 1-79. doi:10.1002/aris.2009.1440430117

Bass, F. M. (1969). A new product growth for model consumer durables. *Management Science, 15*(5), 215-227.

Berger, N., Borgs, C., Chayes, J. T., & Saberi, A. (2005, 2005). *On the Spread of Viruses on the Internet.* Paper presented at the Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms, Vancouver, British Columbia.

Borgatti, S. P., & Halgin, D. S. (2011). On Network Theory. *Organization Science, 22*(5), 1168-1181. doi:10.1287/orsc.1100.0641

Brass, K. (2002). Pushing e-learning. *Sales and Marketing Management, 154*(3), 56.

Brunswicker, S., Matei, S., Zentner, A., Zentner, M., L., & Klimeck, G. (2015). *Creating Impact in the Digital Space: Digital Practice Dependency in Developer Communities of Digital Innovation in Science*. Working Paper. Research Center for Open Digital Innovation. West-Lafayette. Purdue University.

Bulte, C. V. d., & Lilien, G. L. (2001). Medical Innovation Revisited: Social Contagion versus Marketing Efforts. *American Journal of Sociology, 106*(5), 1409-1435. Retrieved from http://www.jstor.org/stable/10.1086/320819 .

Burt, R. S. (1992). *Structural Holes: The Social Structure of Competition.*: Harvard University Press, Cambridge, MA.

Burt, R. S. (2004). Structural Holes and Good Ideas. *American Journal of Sociology, 110*(2), 349-399. Retrieved from http://www.jstor.org/stable/10.1086/421787

Cambridge, U. o. (2010). Computer Laboratory: What is Subversion? Retrieved from https://www.cl.cam.ac.uk/local/web/subversion/introduction/subversionguide.html

Case, A. C., & Rosen, H. S. (1993). Budget spillovers and fiscal policy interdependence: Evidence from the states. *Journal of Public Economics, 52*, 285-307.

Chandrasekhar, A., & Jackson, M. (2013). Tractable and consistent random graph models. *Available at SSRN 2150428*.

Chatterjee, S., & Diaconis, P. (2013). Estimating and understanding exponential random graph models. *The Annals of Statistics, 41*(5), 2428-2461. doi:10.1214/13-aos1155

Coleman, J. S. (1998). Social Capital in the Creation of Human Capital. *American Journal of Sociology, 94*(Supplement: Organizations and Institutions: Sociological and Economic Approaches to the Analysis of Social Structure), S95-S120. Retrieved from http://www.jstor.org/stable/2780243

Constant, D., Sproull, L., & Kiesler, S. (1996). The Kindness of Strangers: The Usefulness of Electronic Weak Ties for Technical Advice. *Organization Science, 7*, 119-135.

Cooper, C., & Frieze, A. (2003). A general model of web graphs. *Random Struct. Algorithms, 22*, 311-335.

Cropanzano, R., & Mitchell, M. S. (2005). Social Exchange Theory: An Interdisciplinary Review. *Journal of Management, 31*(6), 874-900. doi:10.1177/0149206305279602

Crowston, K., Wei, K., Howison, J., & Wiggins, A. (2012). Free/Libre open-source software development. *ACM Computing Surveys, 44*(2), 1-35. doi:10.1145/2089125.2089127

Daskovska, T., Simar, L., & Bellegem,S. (2010). Forecasting the Malmquist productivity index. *Journal of Productivity Analysis, Springer, vol. 33(2)*, 97–107.

Dasu, T., & Johnson, T. (2003). *Exploratory Data Mining and Data Cleaning*: Wiley.

Delre, S. A., Jager, W., Bijmolt, T. H. A., & Janssen, M. A. (2007). Targeting and timing promotional activities: An agent-based model for the takeoff of new products. *Journal of Business Research, 60*(8), 826-835. doi:10.1016/j.jbusres.2007.02.002

Dubin, R. A. (1998). Spatial Autocorrelation: A Primer. *Journal of Housing Economics, 7*, 304-327.

Eck, v. S. P., Wander Jager, & Leeflang, P. S. H. (2011). Opinion Leaders' Role in Innovation Diffusion: A Simulation Study. *Journal of Product Innovation Management, 28*, 187-203.

Ekeh, P. P. (1974). *Social exchange theory: the two traditions*: Harvard University Press.

Elhorst, J. P., Lacombe, D. J., & Piras, G. (2012). On model specification and parameter space definitions in higher order spatial econometric models. *Regional Science and Urban Economics, 42*(1-2), 211-220. doi:10.1016/j.regsciurbeco.2011.09.003

Erdos, P., & Renyi, A. (1959). On random graphs 1. *Publicationes Mathematicae, 6*, 290-297.

Faraj, S., & Johnson, S. L. (2011). Network Exchange Patterns in Online Communities. *Organization Science, 22*(6), 1464-1480. doi:10.1287/orsc.1100.0600

Fibich, G., & Gibori, R. i. (2010). Aggregate Diffusion Dynamics in Agent-Based Models with a Spatial Structure. *Operations Research, 58*(5), 1450-1468. doi:10.1287/opre.1100.0818

Firth, D. R., Lawrence, C., & Clouse, S. F. (2006). Predicting Internet-based online community size and time to peak membership using the bass model of new product growth. *Interdisciplinary Journal of Information, Knowledge, and Management, 1*(1), 1-12.

Flynn, F. J. (2005). Identity Orientations and Forms of Social Exchange in Organizations. *The Academy of Management Review, 30*, 737-750. doi:10.2307/20159165

Frank, O., & Strauss, D. (1986). Markov Graphs. *Journal of the American Statistical Association, 81*(395), 832-842. Retrieved from http://links.jstor.org/sici?sici=0162-1459%28198609%2981%3A395%3C832%3AMG%3E2.0.CO%3B2-C

Friedkin, N. E. (1991). Theoretical Foundations for Centrality Measures. *American Journal of Sociology, 96*(6), 1478-1504. doi:10.2307/2781908

Friedrich, S., & Mark, T. (1998). A Kolmogorov-type test for second-order stochastic dominance. *Statistics & Probability Letters, 37*, 183-193.

Fulk, J., Heino, R., Flanagin, A. J., Monge, P. R., & Bar, F. (2004). A Test of the Individual Action Model for Organizational Information Commons. *Organization Science, 15*, 569-585.

Gabor, C., & Nepusz, T. (2006). The igraph software package for complex network research.

Gawer, A., & Cusumano, M. A. (2014). Industry Platforms and Ecosystem Innovation. *Journal of Product Innovation Management, 31*(3), 417-433. doi:10.1111/jpim.12105

Gold, A. (2007). Cyberinfrastructure, Data, and Libraries, Part 1. *D-Lib Magazine, 13*(9/10).

Gonzalez-Brambila, C. N., Veloso, F. M., & Krackhardt, D. (2013). The impact of network embeddedness on research output. *Research Policy, 42*(9), 1555-1567. doi:10.1016/j.respol.2013.07.008

Granovetter, M. S. (1973). The Strength of Weak Ties. *American Journal of Sociology, 78*(6), 1360-1380.

Greene, W. H. (2003). *Econometric analysis* (Vol. 5): Prentice hall.

Guo, Z. (2012). Stochastic Dominance and Its Applications in Portfolio Management. Retrieved from http://nbn-resolving.de/urn:nbn:de:bsz:352-242222

Hansen, M. T. (1999). The Search-Transfer Problem: The Role of Weak Ties in Sharing Knowledge across organization Subunits. *Administrative Science Quarterly, 44*(1), 82-111. Retrieved from http://www.jstor.org/stable/2667032

Hargadon, A., & Sutton, R. I. (1997). Technology Brokering and Innovation in a Product Development Firm. *Administrative Science Quarterly, 42*(4), 716-749. Retrieved from http://www.jstor.org/stable/2393655

Hargadon, A. B. (2002). BROKERING KNOWLEDGE: LINKING LEARNING AND INNOVATION. *Research in Organizational Behavior, 24*, 41-85.

Holland, P. W., & Leinhardt, S. (1981). An Exponential Family of Probability Distributions for Directed Graphs. *Journal of the American Statistical Association, 76*(373), 33-50. Retrieved from http://links.jstor.org/sici?sici=0162-1459%28198103%2976%3A373%3C33%3AAEFOPD%3E2.0.CO%3B2-Q

Holtgrewe, U. (2004). Articulating the Speed(s) of the Internet: The Case of Open Source/Free Software. *Time & Society, 13*(1), 129-146. doi:10.1177/0961463x04040750

Hoskin, T. *Parametric and non-parametric: Demystifying the terms.* Mayo Clinic CTSA BERD Resource. Retrieved from http://www.mayo.edu/mayo-edudocs/center-for-translational-science-activities-documents/berd-5-6.pdf.

Hunter, D. R., Goodreau, S. M., & Handcock, M. S. (2008). Goodness of Fit of Social Network Models. *Journal of the American Statistical Association, 103*(481), 248-258. doi:10.1198/016214507000000446

Jackson, M. O. (2008). *Social and Economic Networks*: Princeton University Press.

Jackson, M. O., & Rogers, B. W. (2007a). Meeting Strangers and Friends of Friends: How Random Are Social Networks? *American Economic Review, 97*(3), 890-915. Retrieved from http://www.jstor.org/stable/30035025

Jackson, M. O., & Rogers, B. W. (2007b). Relating Network Structure to Diffusion Properties through Stochastic Dominance. *The B.E. Journal of Theoretical Economics, 7*(1).

Jain, D. C., & Rao, R. C. (1990). Effect of Price on the Demand for Durables: Modeling, Estimation, and Findings. *Journal of Business & Economic Statistics, 8*(2), 163-170. doi:10.2307/1391978

James, S. C. (1988). Social Capital in the Creation of Human Capital. *American Journal of Sociology, 94*. Retrieved from http://www.jstor.org/stable/2780243

Jarvenpaa, S. L., & Leidner, D. E. (1999). Communication and Trust in Global Virtual Teams Organizations. *Organization Science, 10*(6), 791-815. Retrieved from http://www.jstor.org/stable/2640242

Jin, L., Chen, Y., Wang, T., Hui, P., & Vasilakos, A. V. (2013). Understanding User Behavior in Online Social Networks: A Survey. *IEEE Communications Magazine*.

Kanawattanachai, P., & Yoo, Y. (2007). The Impact of Knowledge Coordination on Virtual Team Performance over Time. *MIS Quarterly, 31*(4), 783-808.

Kankanhalli, A., Tan, B. C. Y., & Wei, K.-K. (2005). Contributing Knowledge to Electronic Knowledge Repositories: An Empirical Investigation. *MIS Quarterly, 29*(1), 113-143. Retrieved from http://www.jstor.org/stable/25148670

Kilduff, M., Tsai, W., & Hanke, R. (2006). A Paradigm Too Far? A Dynamic Stability Reconsideration of the Social Network Research Program. *The Academy of Management Review, 31*, 1031-1048. doi:10.2307/20159264

Kitsak, M., Gallos, L. K., Havlin, S., Liljeros, F., Muchnik, L., Stanley, H. E., & Makse, H. A. (2010). Identification of influential spreaders in complex networks. *Nature Physics, 6*(11), 888-893. doi:10.1038/nphys1746

Klimeck, G., McLennan, M., Brophy, S. B., Adams-III, G. B., & Lundstrom, M. S. (2008). nanoHUB.org: Advancing Education and Research in Nanotechnology. *Computing in Science& Engineering (IEEE Computer Society), 10*(5). Retrieved from doi:10.1109/MCSE.2008.120

Kling, R., McKim, G., & King, A. (2003). A Bit More to It: Scholarly Communication Forums as Socio-Technical Interaction Networks. *Journal of the American Society for Information Science and Technology, 54*(1), 47-67.

Kumar, R., Raghavan, P., Rajagopalan, S., Sivakumar, D., Tomkins, A., & Upfal, E. (2000, 2000). *Stochastic models for the web graph*.

Laciana, C. E., Rovere, S. L., & Podestá, G. P. (2013). Exploring associations between micro-level models of innovation diffusion and emerging macro-level adoption patterns. *Physica A: Statistical Mechanics and its Applications, 392*(8), 1873-1884. doi:10.1016/j.physa.2012.12.023

Lawrence, K. D., & Lawrton, W. H. (1981). Applications of Diffusion Models: Some Emperical Results. In Y. Wind, V. Mahajan, & R. Cardozo (Eds.), *New Product Forecasting* (pp. 529-541). Lexington, MA: Lexington Books.

Lee, L.-f., & Liu, X. (2009). Efficient Gmm Estimation of High Order Spatial Autoregressive Models with Autoregressive Disturbances. *Econometric Theory, 26*(01), 187. doi:10.1017/s0266466609090653

Leenders, R. T. A. J. (2002). Modeling social influence through network autocorrelation: constructing the weight matrix. *Social Networks, 24*, 21-47.

LeSage, J., & Pace, K. R. (2009). *Introduction to Spatial Econometrics*: CRC Press.

LeSage, J. P., & Pace, R. K. (2011). Pitfalls in Higher Order Model Extensions of Basic Spatial Regression Methodology. *The Review of Regional Studies, 41*(1), 13-26.

Levine, S. S., & Prietula, J. M. (2015). Open Collaboration for Innovation: Principles and Performance. *Organization Science, Forthcoming*.

Li, E. Y., Liao, C. H., & Yen, H. R. (2013). Co-authorship networks and research impact: A social capital perspective. *Research Policy, 42*(9), 1515-1530. doi:10.1016/j.respol.2013.06.012

Lusher, D., Koskinen, J., & Robbins, G. (2013). *Exponential random graph models for social networks : theories, methods, and applications*: Cambridge New York : Cambridge University Press.

Macke, J., & Dilly, E. K. (2010). Social Capital Dimensions in Collaborative Networks: The Role Of Linking Social Capital. *International Journal of Social Inquiry, 3*(2), 121-136.

Mahajan, V., Muller, E., & Bass, F. M. (1990). New Product Diffusion Models in Marketing: A Review and Directions for Research. *Journal of Marketing, 54*(1), 1-26. Retrieved from http://www.jstor.org/stable/1252170

Marsaglia, G., Tsang, W. W., & Wang, J. (2003). Evaluating Kolmogorov\'s Distribution. *2003, 8*(18), 4. doi:10.18637/jss.v008.i18

Matei, S. A. (2014).A social network analysis and entropy approach to defining and measuring the impact of social roles on social media and in science collaboration platforms – a case study with nanoHUB.org (Exploratory Social Science Grant, Office of the Vice President for Research, Purdue University, PI: Sorin Adam Matei, http://kredible.net/in/?p=647)

Matei, S. A., Bertino, E., Zhu, M., Liu, C., Si, L., & Britt, B. (2015). A Research Agenda for the Study of Entropic Social Structural Evolution, Functional Roles, Adhocratic Leadership Styles, and Credibility in Online Organizations and Knowledge Markets. In E. Bertino & S. A. Matei (Eds.), *Roles, Trust, and Reputation in Social Media Knowledge Markets* (pp. 3–33). Cham: Springer International Publishing. Retrieved from http://springer.libdl.ir/chapter/10.1007/978-3-319-05467-4_1

McFadyen, M. A., & Albert, A. C. J. (2004). Social Capital and Knowledge Creation: Diminishing Returns of the Number and Strength of Exchange. *The Academy of Management Journal, 47*(5), 735-746. Retrieved from http://www.jstor.org/stable/20159615

McLennan, M. (2012). Rappture Bootcamp: Building and Deploying Tools.

McLure, W. M., & Faraj, S. (2005). Why Should I Share? Examining Social Capital and Knowledge Contribution in Electronic Networks of Practice. *MIS Quarterly, 29*(1), 35-57. Retrieved from http://www.jstor.org/stable/25148667

Meade, N., & Islam, T. (2006). Modelling and forecasting the diffusion of innovation – A 25-year review. *International Journal of Forecasting, 22*(3), 519-545. doi:10.1016/j.ijforecast.2006.01.005

Monge, P. R., & Contractor, N. S. (2003). *Theories of communication networks*: Oxford University Press.

Moody, J., & White, T. D. (2003). Structural Cohesion and Embeddedness: A Hierarchical Concept of Social Groups. *American Sociological Review, 68*(1), 103-127. Retrieved from http://www.jstor.org/stable/3088904

nanoHUB. (2014). About US. Retrieved from https://nanohub.org/about

Newman, M. E. J. (2003). Mixing patterns in networks. *Physical Review E, 67*(2). doi:10.1103/PhysRevE.67.026126

O'Malley, A. J., & Marsden, P. V. (2008). The Analysis of Social Networks. *Health Serv Outcomes Res Methodol, 8*(4), 222-269. doi:10.1007/s10742-008-0041-z

Obstfeld, D. (2005). Social Networks, the Tertius Iungens Orientation, and Involvement in Innovation. *Administrative Science Quarterly, 50*(1), 100-130. Retrieved from http://www.jstor.org/stable/30037177

Orlikowski, W. J. (2000). Using Technology and Constituting Structures: A Practice Lens for Studying Technology in Organizations. *Organization Science, 11*(4), 404-428. Retrieved from http://www.jstor.org/stable/2640412

Páez, A., Scott, D. M., & Volz, E. (2008). Weight matrices for social influence analysis: An investigation of measurement errors and their effect on model identification and estimation quality. *Social Networks, 30*(4), 309-317. doi:10.1016/j.socnet.2008.05.001

Peddibhotla, N. B., & Subramani, M. R. (2007). Contributing to Public Document Repositories: A Critical Mass Theory Perspective. *Organization Studies, 28*(3), 327-346. doi:10.1177/0170840607076002

Pennock, D. M., Flake, G. W., Lawrence, S., Glover, E. J., & Giles, C. L. (2002). Winners don't take all: Characterizing the competition for links on the web. *Proceedings of the National Academy of Sciences, 99*(8), 5207-5211. doi:10.1073/pnas.032085699

Peres, R., Muller, E., & Mahajan, V. (2010). Innovation diffusion and new product growth models: A critical review and research directions. *International Journal of Research in Marketing, 27*(2), 91-106. doi:10.1016/j.ijresmar.2009.12.012

Recker, J. (2013). *Scientific Research in Information Systems*: Springer Heidelberg New York Dordrecht London.

Ringquist, E. (2013). *Meta-Analysis for Public Management and Policy*: Wiley.

Robins, G., Pattison, P., Kalish, Y., & Lusher, D. (2007). An introduction to exponential random graph (p*) models for social networks. *Social Networks, 29*(2), 173-191. doi:10.1016/j.socnet.2006.08.002

Rogers, E. M. (1983). *Diffusion of Innovations*. University of Illinois at Urbana-Champaign's Academy for Entrepreneurial Leadership Historical Research Reference in Entrepreneurship.

Rossi, M. A. (2006). 2 - Decoding the Free/Open Source Software Puzzle: A Survey of Theoretical and Empirical Contributions. In J. B. J. H. Schröder (Ed.), *The Economics of Open Source Software Development* (pp. 15-55). Amsterdam: Elsevier.

Rullani, F., & Haefliger, S. (2013). The periphery on stage: The intra-organizational dynamics in online communities of creation. *Research Policy, 42*(4), 941-953. doi:10.1016/j.respol.2012.10.008

Scacchi, W. (2007). Free/Open Source Software Development: Recent Research Results and Methods. *69*, 243-295. doi:10.1016/s0065-2458(06)69005-0

Schmittlein, D. C., & Mahajan, V. (1982). Maximum Likelihood Estimation for an Innovation Diffusion Model of New Product Acceptance. *Marketing Science, 1*(1), 57-78. doi:doi:10.1287/mksc.1.1.57

Schroeder, R., Jennifer, A. d., & Jenny, F. (2007). *e-Research Infrastructures and Scientific Communication.* Paper presented at the IATUL Conferences.

Schumpeter, J. A. (1942). *Capitalism, Socialism, and Democracy*. University of Illinois at Urbana-Champaign's

Shanahan, C. J., Hooker, N. H., & Sporleder, T. L. (2008). The diffusion of organic food products: toward a theory of adoption. *Agribusiness, 24*(3), 369-387. doi:10.1002/agr.20164

Singh, J. (2007). *External Collaboration, Social Networks and Knowledge Creation: Evidence from Scientific Publications*. Paper presented at the DRUID Summer Conference 2007, Copenhagen, CBS, Denmark.

Snijders, T. A. B., Pattison, P. E., Robins, G. L., & Handcock, M. S. (2006). NEW SPECIFICATIONS FOR EXPONENTIAL RANDOM GRAPH MODELS. *Sociological Methodology, 36*, 99-153. doi:10.1111/j.1467-9531.2006.00176.x

Søndergaard, F. T., Andersen, J., & Hjørland, B. (2003). Documents and the communication of scientific and scholarly information. *Journal of Documentation, 59*(3), 278-320. doi:10.1108/00220410310472509

Srinivasan, V., & Mason, C. H. (1986). Nonlinear Least Squares Estimation of New Product Diffusion Models. *Marketing Science, 5*(2), 169-178. Retrieved from http://www.jstor.org.ezproxy.lib.purdue.edu/stable/183671

Stanley, T. D., & Doucouliagos, H. (2012). *Meta-regression Analysis in Economics and Business*: Routledge.

Stewart, C. A., Simms, S., Plale, B., Link, M., Hancock, D. Y., & Fox, G. C. (2010). *What is Cyberinfrastructure?* Paper presented at the 38th Annual Fall Conference on SIGUCCS, New York.

Susarla, A., Oh, J.-H., & Tan, Y. (2012). Social Networks and the Diffusion of User-Generated Content: Evidence from YouTube. *Information Systems Research, 23*(1), 23-41. doi:10.1287/isre.1100.0339

Tiefelsdorf, M. (2000). *Modelling Spatial Processes: The Identification and Analysis of Spatial Relationships in Regression Residuals by Means of Moran's I.*

Uzzi, B. (1997). Social Structure and Competition in Interfirm Networks: The Paradox of Embeddedness. *Administrative Science Quarterly, 42*(1), 35-67. Retrieved from http://www.jstor.org/stable/2393808

Valente, T. W., Kathryn, C., Lakon, C., & Costenbader, E. (2008). How correlated are network centrality measures? *Connect (Tor), 28*(1), 16-26.

Van den Bulte, C., & Stremersch, S. (2004). Social Contagion and Income Heterogeneity in New Product Diffusion: A Meta-Analytic Test. *Marketing Science, 23*(4), 530-544. doi:10.1287/mksc.1040.0054

Vass, B. (2007, 2007). *Migrating to open source: Have no fear*.

Vázquez, A. (2003). Growing network with local rules: Preferential attachment, clustering hierarchy, and degree correlations. *Physical Review E, 67*(5). doi:10.1103/PhysRevE.67.056104

Wasko, M. M., & Faraj, S. (2000). "It is what one does": why people participate and help others in electronic communities of practice. *Journal of Strategic Information Systems, 9*, 155-173.

Wasserman, S., & Faust, K. (1994). *Social network analysis: Methods and applications* (Vol. 8): Cambridge university press.

Wasserman, S., & Pattison, P. (1996). Logit Models and Logistic Regressions for Social Networks: I. An Introducton to Markov Graphs and p*. *Psychometrika, 61*(3), 401-425.

Watts, D. J. (1999). *Small worlds: the dynamics of networks between order and randomness*: Princeton university press.

Winfree, R., Dushoff, J., Crone, E. E., Schultz, C. B., Budny, R. V., Williams, N. M., & Kremen, C. (2005). Testing simple indices of habitat proximity. *The American Naturalist, 165*(6), 707-717.

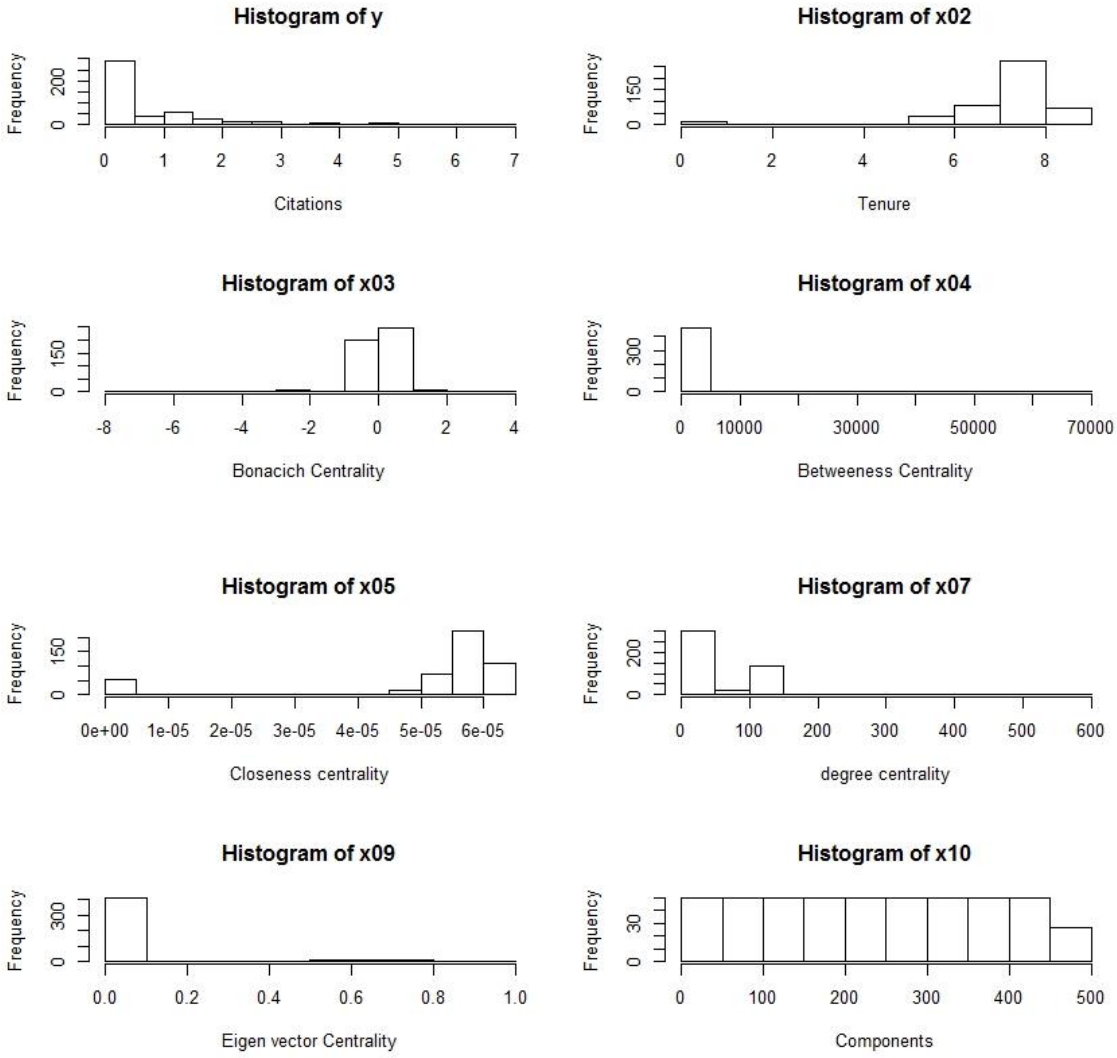
Wooldridge, J. (2008). *Introductory Econometrics: A Modern Approach*: Cengage Learning.

Wright, M., Upritchard, C., & Lewis, T. (1997). A Validation of the Bass New Product Diffusion Model in New Zealand. *Marketing Bulletin, 8*(2), 15-29. Retrieved from http://marketing-bulletin.massey.ac.nz

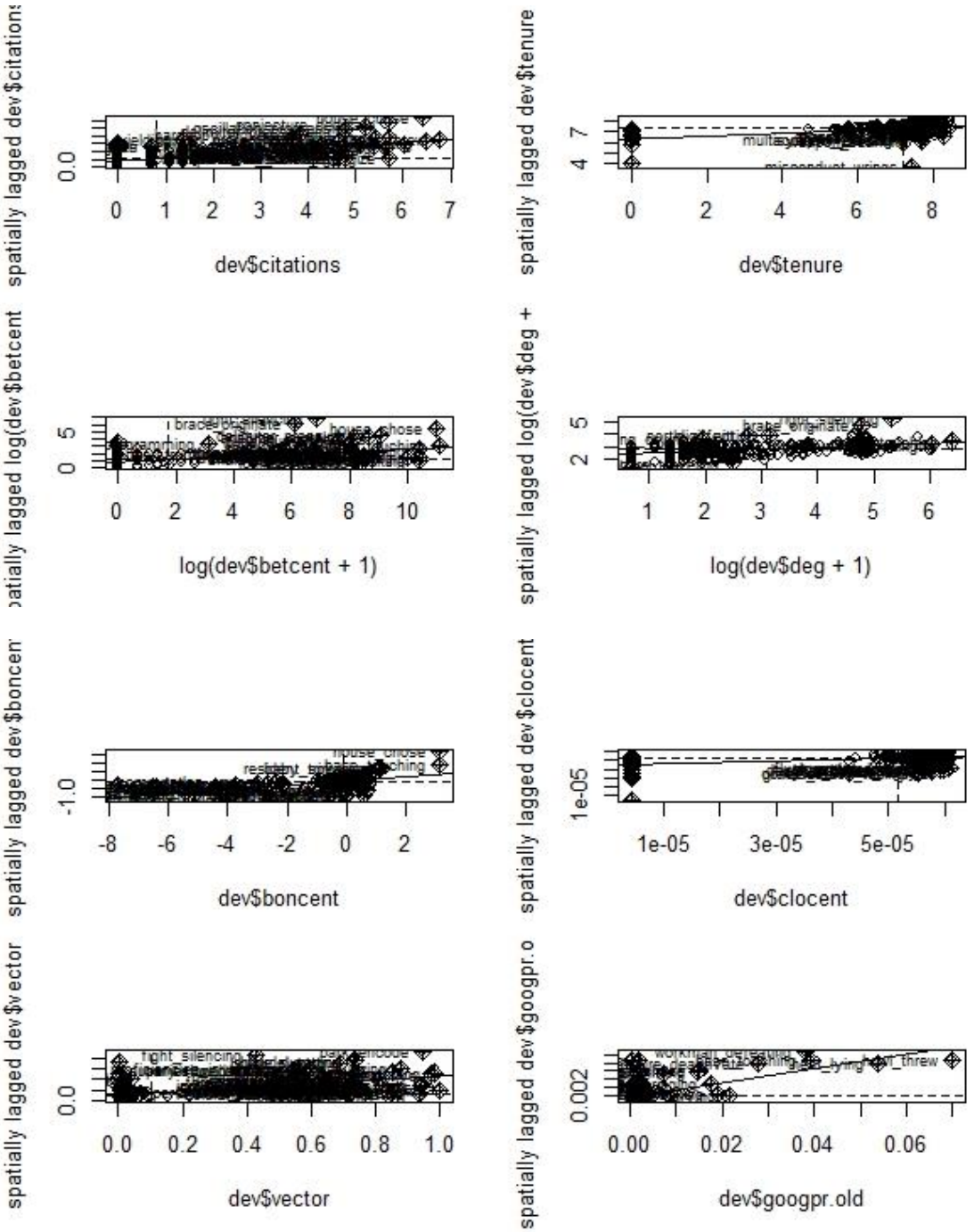
Yoo, Y., Boland, R. J., Lyytinen, K., & Majchrzak, A. (2012). Organizing for Innovation in the Digitized World. *Organization Science, 23*(5), 1398-1408. doi:10.1287/orsc.1120.0771

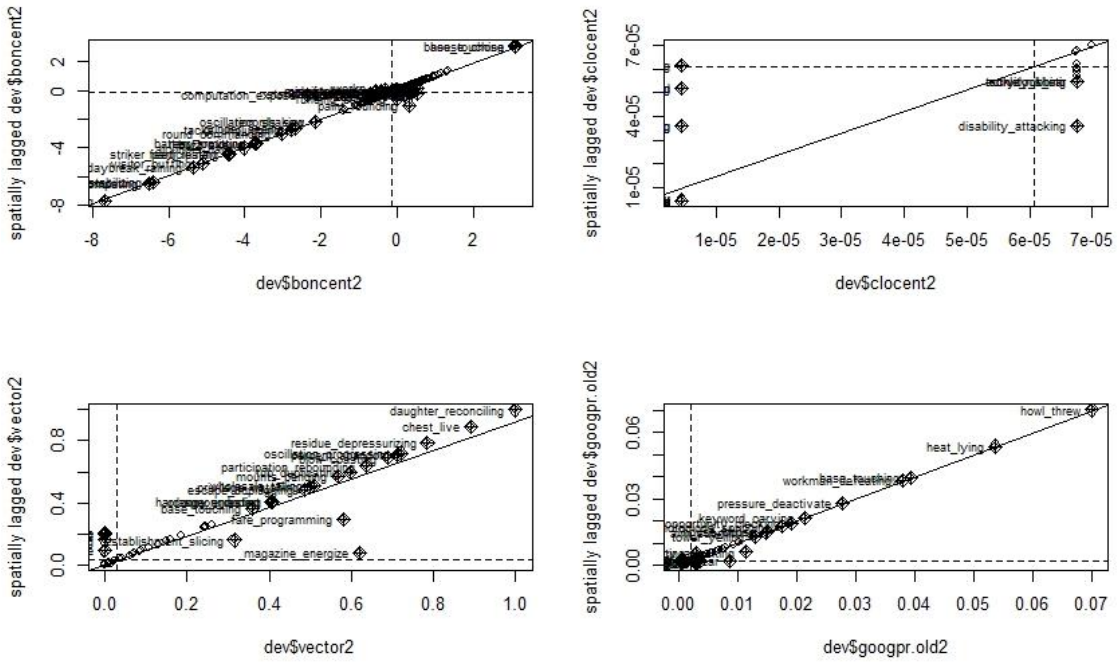
APPENDICES

Appendix A Histograms for Endogenous and Predictor Variables

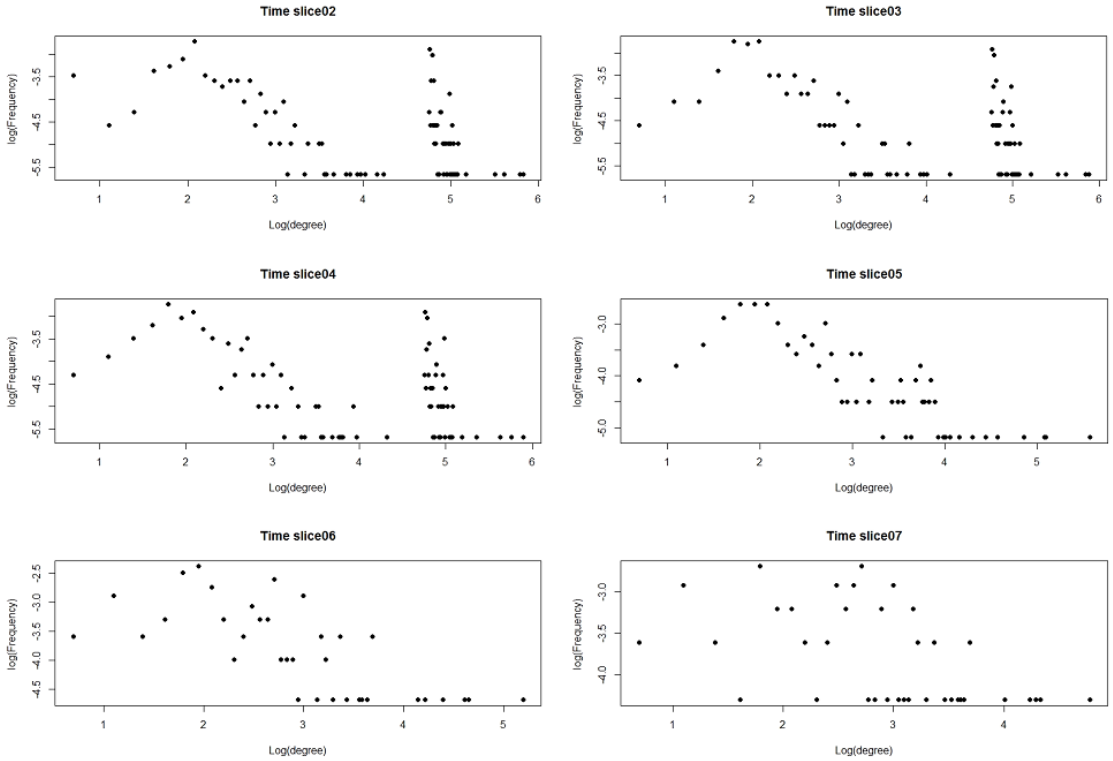


Appendix B Moran's I Scatter Plots for the Endogenous and Predictor Variables




spatially lagged dev$boncent2
dev$boncent2
spatially lagged dev$clocent2
dev$clocent2
spatially lagged dev$vector2
dev$vector2
daughter_reconciling
chest_live
residue_depressurizing
fare_programming
magazine_energize
spatially lagged dev$googpr.old2
dev$googpr.old2
howl_threw
heat_lying
pressure_deactivate
disability_attacking

Appendix C Degree Distributions of the 7-Time Slices



Appendix D R-Codes

Paper 1. R-Codes

```
# Author. Philip Munyua.


#Probit, Spatial Probit, Interaction Models and SDEM models
#Outline
# 1. Data mining (Data Extraction from SVN logs, Data Cleaning and Data Merging,)
# 2. Statistical Analysis (Model Variables Extraction and Visualization, Cleaning and
Model Analysis)

## Data Mining
#Betweeness Matching
#set directory and read files
rm(list=ls())  #Clear working directory
#--load libraries
library(spdep)
library(Matrix)
library(igraph)
library(lmtest)
library(sphet)
library(AER)
library(spatstat)
library(spatialprobit)
library(McSpatial)
library(mfx)
library(stats)
library(Hmisc)
library(utils)
library(Zelig)

# Set/Load  working directory
setwd("C:\\Users\\Philipmunyua\\Google Drive\\nanoHUB research\\02 analysis\\04 develop
level _test spatial\\02 input data")

##Generating Developers Weight Matrix using Gravity Model

# load data into dataframe weight.dat
weight.dat<-read.csv("crystal_viewer_metrics_DiD.csv")

# List of unique tool developers in crystal_names
# Steps to get unique values 1. Get all the unique values 2. Convert them to character 3.
Add them to the vertex_attrib files
crystal_names<-as.character( unique (weight.dat$username) )

# Add the vertex names to the vertex_names vector which will be made a column in the
vertex_attrib dataframe

Names<-crystal_names

# Getting all the vertices (minus) duplicates
Names<-union(vertex_attrib$Names,Names)
vertex_attrib<-data.frame(Names)

# size of the adjacency matrix for crystal ( Just calculate this as we might need it
later on)
crystal_size<-length(crystal_names)

# Number of revisions in total
crystal_rev<-length(crystal[,1])   # could even think of using crystal_rev<-
max(crystal$rev)

# Creating the contribution <crbn> column  to be changed as per Gravitational model
```

```
crbn<-vector(mode="numeric",length=crystal_rev)
for(i in 1:crystal_rev)
{
  crbn[i]<-max(weight.dat$add[i],weight.dat$del[i]) -
0.5*(min(weight.dat$add[i],weight.dat$del[i])) + weight.dat$chrn[i]; # Gravity model
formula numerator
}

# Add the contribution <crbn> column to the crystal.dat data frame
weight.dat<-data.frame(weight.dat,crbn)

# Creating weights using modified gravitational centrality and adding it to the
edge_attributes files
# IMPORTANT -- Looping is backwards
temp_contrib<-0
temp_weight<-0
new_edge_attrib_row <-c("","",0)
for(i in crystal_rev:2)
{
  currval<-i-1 # value that needs to be passed to j
  temp_contrib = weight.dat$crbn[i]
  for(j in currval:1)
  {
    if(weight.dat$username[i]==weight.dat$username[j])
    {temp_contrib = temp_contrib+weight.dat$crbn[j];}

    else
    {
      distance = weight.dat$rev[j]-weight.dat$rev[i];
      if ( length(edge_attrib$FROM[ edge_attrib$FROM
==as.character(weight.dat$username[i]) &
edge_attrib$TO==as.character(weight.dat$username[j])])==0 )
      {  # Create a new edge link between the two
        new_edge_attrib_row = c(  as.character(weight.dat$username[i]) ,
as.character(weight.dat$username[j]) , as.numeric(temp_contrib/( distance^2 ))  );
        edge_attrib<-rbind(edge_attrib,new_edge_attrib_row);   next;
      }
      else {
        temp_weight <- as.numeric(edge_attrib$WEIGHT[edge_attrib$FROM ==
as.character(weight.dat$username[i]) & edge_attrib$TO==
as.character(weight.dat$username[j] )  ]) +  (temp_contrib*tool$crbn[j])/( distance^2 ) ;
        edge_attrib$WEIGHT[edge_attrib$FROM == as.character(weight.dat$username[i]) &
edge_attrib$TO== as.character( weight.dat$username[j] )  ]<-temp_weight
      }
    }
  }
}

#----------Generating Authors Edgelist
aut1<-read.csv("aut1.csv")
dev1<-read.csv("dev1.csv")
aut2<-aut1[, c(3,9)]
dev2<-dev1[, c(4,29)]
write.csv(aut2, file="C:\\Users\\pmunyua\\Google Drive\\nanoHUB research\\02 analysis\\04
develop level _test spatial\\02 input data\\aut2.csv")
write.csv(dev2, file="C:\\Users\\pmunyua\\Google Drive\\nanoHUB research\\02 analysis\\04
develop level _test spatial\\02 input data\\dev2.csv")

#Matching Authors weighted Edgelist

autedge<-read.csv("weightedEdgeList.csv")

for(i in 1:dim(autedge)[1]){

  index2<-which(as.character(autedge$FROM[i])==as.character(aut2$auth_nano_uid))
  if (length(index2)>0){
    autedge[i,"FROM1"]<-aut2$username[index2]
  }else{
    autedge[i,"FROM1"]<-NA
```

```
  }
}

for(i in 1:dim(autedge)[1]){

  index3<-which(as.character(autedge$TO[i])==as.character(aut2$auth_nano_uid))
  if (length(index3)>0){
    autedge[i,"TO1"]<-aut2$username[index3]
  }else{
    autedge[i,"TO1"]<-NA
  }
}

autedge1<-autedge[, c(2,7,3,8,4)]
write.csv(autedge1, file="C:\\Users\\pmunyua\\Google Drive\\nanoHUB research\\02
analysis\\04 develop level _test spatial\\02 input data\\autedge1.csv")

devedge<-read.csv("edge_list_complete_log10.csv")
autedge2<-read.csv("autedge1.csv")

for(i in 1:dim(devedge)[1]) {
    index4 <- which(as.character(devedge$FROM[i])==as.character(autedge2$FROM1)&
                      as.character(devedge$TO[i])==as.character(autedge2$TO1))
          devedge[i,"WEIGHT1"]<-ifelse(length(index4)>0, autedge2$Wgt[index4],0)
     }
write.csv(devedge, file="C:\\Users\\pmunyua\\Google Drive\\nanoHUB research\\02
analysis\\04 develop level _test spatial\\02 input data\\devedge.csv")


##---------------------Calculating Developers (W1) and Authorship (W2) Weight Matrices

#W1
graph_list<-read.csv("edge_list_complete_log10.csv") ## read developers edgelist (the
weight have been shifted by 1 and logged)
graph_list$X<-NULL   #deleting empty
mat<-as.matrix(graph_list[,1:2])

## Make an directed graph
graph<-graph.edgelist(mat,directed=TRUE)
E(graph)$weight<-graph_list[,3]
gmat<-get.adjacency(graph,attr="weight")
#plot(gmat, layout = layout.fruchterman.reingold, vertex.label.family = "sans",
vertex.size = 477, vertex.label = "")
adjmat<-as.matrix(gmat)

for(i in 1:477)
{
  adjmat2[i,i]= 1.0
}
W1<-as.matrix(adjmat)
nb<-mat2listw(adjmat)
## Row standardized weigth matrix:Row standardization creates proportional
##weights in cases where features have an unequal number of neighbors
nb1<-nb2listw(nb$neighbours, style="W")

#W2
graph_list2<-read.csv("devedge.csv") ## logged edge_list
graph_list2$X.1<-NULL
graph_list2$X<-NULL
graph_list2$WEIGHT<-NULL
graph_list2$WGT1<-NULL
mat2<-as.matrix(graph_list2[,1:2])

## Make an directed graph
graph2<-graph.edgelist(mat2,directed=TRUE)
E(graph2)$weight<-graph_list2[,3]
gmat2<-get.adjacency(graph2,attr="weight")
```

```
##-----------Generating Descriptive Statistics and Extracting degree centrality measures
variables
#--------------Assortivity
V(graph2)$foo <- sample(1:3, replace=TRUE, vcount(graph2))
assort<-assortativity.nominal(graph2, types=V(graph2)$foo)
assortivity<-as.data.frame(assort) # set the value as data frame

#clusters
#calculates the "maximal (weakly or strongly) connected components of a graph"
isclus<-is.connected(graph2, mode=c("weak", "strong")) #decided whether the graph is
weakly or strongly connected

#Diameter
#calculates the "length of the longest geodesic"
getdiam<-get.diameter(graph2, directed=TRUE, unconnected=TRUE, weights=NULL) # retunrs a
path with actual diameter
getdiam2<-as.data.frame(getdiam)
farnodes<-farthest.nodes(graph2, directed=TRUE, unconnected=TRUE, weights=NULL) # returns
two vertex ids
farnodes2<-as.data.frame(farnodes)

#Dyad Census (p.85)
dyads<-dyad.census(graph2)
dyads2<-as.data.frame(dyads)

#Graph density
#Density "is the ratio of the number of edges (links) and the number of possible edges"
density<-graph.density(graph2, loops=FALSE)
density2<-as.data.frame(density)

#Average nearest neighbor degree
# "calculates  the average nearest neighbor degree of the given vertices and the same
quantity in the function of the vertex degree"
avneigh<-graph.knn(graph2, vids=V(graph2),weights=NULL)
avneigh1<-as.data.frame(avneigh)

#Reciprocity of graphs
recipro<-reciprocity(graph2, ignore.loops=TRUE, mode=c("default", "ratio"))
recipro2<-as.data.frame(recipro)

#Shortest Path
#Calculates the shortest paths between vertices
shortpath<-shortest.paths(graph2, v=V(graph2),mode=c("all","out","in"),
            weights=NULL, algorithm=c("automatic","unweighted","dijkstra","bellman-
ford","johnson"))
getshortpath<-get.shortest.paths(graph2, 2, to=V(graph2), mode = c("out", "all",
              "in"), weights = NULL, output=c("vpath", "epath", "both"), predecessors =
FALSE, inbound.edges = FALSE)
getallshortpath<-get.all.shortest.paths(graph2, 2, to = V(graph2), mode = c("out",
                                    "all", "in"), weights=NULL)
avshortpath<-average.path.length(graph2, directed=TRUE, unconnected=TRUE)
avshortpath2<-as.data.frame(avshortpath)
pathlenngthhist<-path.length.hist (graph2, directed = TRUE)

#Transivity or clustering coefficient
#A measure of the probability that the adjacency nodes of a node are connected (also
known as clustering coefficient)
clustcoeff<-transitivity(graph2, type=c("undirected", "global", "globalundirected",
                                        "localundirected", "local", "average",
"localaverage",
                                        "localaverageundirected", "barrat", "weighted"),
vids=NULL,
                         weights=NULL, isolates=c("NaN", "zero"))
clustcoeff2<-as.data.frame(clustcoeff)

#Triad Census
triads<-triad.census(graph2)
tri2<-as.data.frame(triads)
```

```
#----Centrality measures (Bonacich, Betweeness, Closeness, Degree, Eigen Vector and
Google Page Rank )
#-------------"Bonacich Power Centrality Scores of Network Positions"
boncent<-bonpow(graph2, nodes=V(graph2), loops=FALSE, exponent=0.1, rescale=FALSE,
tol=1e-7, sparse=TRUE)
BC2<-as.data.frame(boncent) # Set the value as data frame

# Betweeness Centrality
betcent<-betweenness(graph2, v=V(graph2), directed=TRUE, weights=NA, nobigint=TRUE,
normalized=FALSE) # calculates nodes betweenness centrality
betcent.est<-edge.betweenness(graph2, vids=V(graph2), directed=TRUE, cutoff, weights=NA,
nobigint=TRUE) # calculates nodes betweenness centrality with cuttoff paths
BeC2<-as.data.frame(betcent)  # Set the value as data frame

#Closeness Centrality
clocent<-closeness(graph2, vids=V(graph2), mode=c("out", "in", "all","total"),
weights=NULL, normalized=FALSE)
CC2<-as.data.frame(clocent)  # Set the value as data frame

#Degree Centrality
deg<-degree(graph2, v=V(graph2),
mode=c("all","out","in","total"),loops=TRUE,normalized=FALSE)
dC2<-as.data.frame(deg)

#Eigen Vector Centrality
evcent<-evcent(graph2, directed=FALSE, scale=TRUE,weights=NULL,
options=igraph.arpack.default)
evcent2<-as.data.frame(evcent[1])

#google Page Rank
#Calculating google page rank
googpr<-page.rank(graph2, algo=c("prpack","arpack","power"),vids=V(graph2),
directed=TRUE, damping=0.85, personalized=NULL, weights=NULL, options=NULL)
googpr.old<-page.rank.old(graph2, vids=V(graph2), directed=TRUE, niter=1000, eps=0.001,
damping=0.85, old=FALSE)
googpr.old2<-as.data.frame(googpr.old)

#compling vector of new variables
object2<-data.frame(BC2,BeC2,CC2, dC2,evcent2, dcomp2, googpr.old2)
file_cc<-file("C:\\Users\\pmunyua\\Google Drive\\nanoHUB research\\02 analysis\\04
develop level _test spatial\\02 input data\\centrality0130W2.csv","w")
write.csv(object2, file_cc)
close(file_cc)

#Merging data sets
object3<-read.csv("centrality0130W2.csv",header=TRUE)
colnames(object3)[1] <- "Dev_Name" #Renaming a column
class(object3)
newdev2<-merge(dev, object3, by="Dev_Name") #Merging developer network with the new
variables by the common variable "Dev Name"
newdev2$X<-NULL
class(newdev2)

file_dev2<-file("C:\\Users\\pmunyua\\Google Drive\\nanoHUB research\\02 analysis\\04
develop level _test spatial\\02 input data\\dev_attrib_ver0130W2.csv","w")
write.csv(newdev2, file_dev2)
close(file_dev2)

#-------------Models

##SDEM with Plots
### Converting from Matrix to ListW object for spatial regression ###

## set the working directory to Nanohub
rm(list=ls())
setwd("C:\\Users\\Philipmunyua\\Google Drive\\nanoHUB research\\02 analysis\\04 develop
level _test spatial\\02 input data")

library(spdep)
```

```
library(Matrix)
library(igraph)
library(lmtest)
library(sphet)
library(AER)
library(spatstat)
library(spatialprobit)
library(McSpatial)
library(mfx)
library(stats)
library(Hmisc)
library(utils)
library(Zelig)
library(lme4)
library(foreign)
library(nlme)
library(matrixcalc)

#---LOAD DEVELOPER ATTRIBUTES VARIABLES TABLE IN DEV DATA FRAME
dev<-read.csv("dev_attrib_ver0130w1w2.csv",header=TRUE)

##Converting data to log after adding (+1)
dev$citations=log(dev$citations+1)
class(dev$citations)
dev$tenure=log(dev$tenure+1)


#dev$dummy_type=log(dev$dummy_type+1)
#Data extraction  for regressions
data<-as.matrix(dev)
#DV
y=as.numeric(subset(data,select=c(citations)))

#IVs
x01=rep(c(1), 477)
x02=as.numeric(subset(data,select=c(tenure)))
x03=as.numeric(subset(data,select=c(boncent)))
x04=as.numeric(subset(data,select=c(betcent)))
x05=as.numeric(subset(data, select=c(clocent)))
x06=as.numeric(subset(data, select=c(dummy_type)))
x07=as.numeric(subset(data, select=c(deg)))
x09=as.numeric(subset(data, select=c(vector)))
x10=as.numeric(subset(data, select=c(comp)))
x11=as.numeric(subset(data, select=c(googpr.old)))
xx03=as.numeric(subset(data,select=c(boncent2)))
xx04=as.numeric(subset(data,select=c(betcent2)))
xx05=as.numeric(subset(data, select=c(clocent2)))
xx07=as.numeric(subset(data, select=c(deg2)))
xx09=as.numeric(subset(data, select=c(vector2)))
xx10=as.numeric(subset(data, select=c(comp2)))
xx11=as.numeric(subset(data, select=c(googpr.old2)))
y=as.matrix(y)

#Histograms plots
attach(mtcars)
par(mfrow=c(2,2))
hist(y, xlab="Citations")
hist(x02, xlab="Tenure")
hist(x03, xlab="Bonacich Centrality")
hist(x04, xlab="Betweeness Centrality")
attach(mtcars)
par(mfrow=c(2,2))
hist(x05, xlab="Closeness centrality")
hist(x07, xlab="degree centrality")
hist(x09, xlab="Eigen vector Centrality")
hist(x10, xlab="Components")
#POwerlaw tests
#Fitting Power Law
#fitting a power-law distribution
```

```
powerlawfit.y<-power.law.fit(y, xmin=NULL, start=2, force.continuous=FALSE,
implementation=c("plfit", "R.mle"))
powerlaw.y<-as.data.frame(powerlawfit.y)
powerlawfit.3<-power.law.fit(x03, xmin=NULL, start=2, force.continuous=FALSE,
implementation=c("plfit", "R.mle"))
powerlaw.3<-as.data.frame(powerlawfit.3)
powerlawfit.4<-power.law.fit(x04, xmin=NULL, start=2, force.continuous=FALSE,
implementation=c("plfit", "R.mle"))
powerlaw.4<-as.data.frame(powerlawfit.4)
powerlawfit.5<-power.law.fit(x05, xmin=NULL, start=2, force.continuous=FALSE,
implementation=c("plfit", "R.mle"))
powerlaw.5<-as.data.frame(powerlawfit.5)
powerlawfit.7<-power.law.fit(x07, xmin=NULL, start=2, force.continuous=FALSE,
implementation=c("plfit", "R.mle"))
powerlaw.7<-as.data.frame(powerlawfit.7)
powerlawfit.9<-power.law.fit(x09, xmin=NULL, start=2, force.continuous=FALSE,
implementation=c("plfit", "R.mle"))
powerlaw.9<-as.data.frame(powerlawfit.9)
powerlawfit.10<-power.law.fit(x10, xmin=NULL, start=2, force.continuous=FALSE,
implementation=c("plfit", "R.mle"))
powerlaw.10<-as.data.frame(powerlawfit.10)

##Correlation
Xy=cbind(y, x02, x03, x04, x05, x06, x07, x09, x10, x11, xx03, xx04, xx05, xx07, xx09,
xx10, xx11)
corr<-rcorr(as.matrix(Xy))
write.table(corr, file="C:\\Users\\pmunyua\\Google Drive\\nanoHUB research\\02
analysis\\04 develop level _test spatial\\02 input data\\corr.txt", sep="\t")
#Summary Statistics
summary(Xy)
## Moran Tests under randomisation and normality
moran.test(y,nb1) # row standardized
moran.test(y,nb1, randomisation=FALSE)
moran.test(x02,nb1) # row standardized
moran.test(x02,nb1, randomisation=FALSE)
moran.test(x03,nb1) # row standardized
moran.test(x03,nb1, randomisation=FALSE)
moran.test(x04,nb1) # row standardized
moran.test(x04,nb1, randomisation=FALSE)
moran.test(x05,nb1) # row standardized
moran.test(x05,nb1, randomisation=FALSE)
moran.test(x07,nb1) # row standardized
moran.test(x07,nb1, randomisation=FALSE)
moran.test(x09,nb1) # row standardized
moran.test(x09,nb1, randomisation=FALSE)
moran.test(x10,nb1) # row standardized
moran.test(x10,nb1, randomisation=FALSE)
moran.test(x11,nb1) # row standardized
moran.test(x11,nb1, randomisation=FALSE)
moran.test(xx03,nb1) # row standardized
moran.test(xx03,nb1, randomisation=FALSE)
moran.test(xx04,nb1) # row standardized
moran.test(xx04,nb1, randomisation=FALSE)
moran.test(xx05,nb1) # row standardized
moran.test(xx05,nb1, randomisation=FALSE)
moran.test(xx07,nb1) # row standardized
moran.test(xx07,nb1, randomisation=FALSE)
moran.test(xx09,nb1) # row standardized
moran.test(xx09,nb1, randomisation=FALSE)
moran.test(xx10,nb1) # row standardized
moran.test(xx10,nb1, randomisation=FALSE)
moran.test(xx11,nb1) # row standardized
moran.test(xx11,nb1, randomisation=FALSE)


##Moran Plots for the depedent and indepedent variable

# 4 figures arranged in 2 rows and 2 columns
attach(mtcars)
```

```
par(mfrow=c(2,2))
moran.plot(dev$citations,nb1)
moran.plot(dev$tenure,nb1)
moran.plot(log(dev$betcent2+1),nb1)
moran.plot(log(dev$deg2+1),nb1)
# 4 figures arranged in 2 rows and 2 columns
attach(mtcars)
par(mfrow=c(2,2))
moran.plot(dev$boncent2,nb1)
moran.plot(dev$clocent2,nb1)
moran.plot(dev$vector2,nb1)
moran.plot(dev$googpr.old2,nb1)

##Getting logs and power functions
x04<-log(1+x04)  #between centrality
x05<-100000*x05  # closeness centrality (Justification for amplification)
x07<-log(1+x07) #degree centrality
x11<-100*x11   #google.page rank
ln_xx04<-ln(xx04+1) # betcent2
xx09    vector2

X=cbind(x02, x03, x04, x05, x07, x09, x10, x11, xx03, xx04, xx05, xx07, xx09, xx10, xx11)
class(X)

#Generating new variables with W1 and W2
W1X<-W1%*%X
W2X<-W2%*%X

#Extracting variables
x02w1<-as.data.frame(W1X[,1])
x03w1<-as.data.frame(W1X[,2])
x04w1<-as.data.frame(W1X[,3])
x05w1<-as.data.frame(W1X[,4])
x07w1<-as.data.frame(W1X[,5])
x09w1<-as.data.frame(W1X[,6])
x10w1<-as.data.frame(W1X[,7])
x11w1<-as.data.frame(W1X[,8])

x02w2<-as.data.frame(W2X[,1])
x03w2<-as.data.frame(W2X[,9])
x04w2<-as.data.frame(W2X[,10])
x05w2<-as.data.frame(W2X[,11])
x07w2<-as.data.frame(W2X[,12])
x09w2<-as.data.frame(W2X[,13])
x10w2<-as.data.frame(W2X[,14])
x11w2<-as.data.frame(W2X[,15])

#Transformations

#x03w1  local1 boncent  normal
x07w1<-log(x07w1)# degree centrality transformation for power law
#x02w2  local2 tenure   normal
#x03w2  local 2 boncent normal
#x04w2  local 2 betcent power law
#x05w2  local2 clocent  normal
#x07w2  local 2 degree  powerlaw
#x10w2  local 2 component   normal
#x11w2  local 2 googlepage rank discard

#x02w2  local2 tenure   normal
#x03w2  local 2 boncent normal
x04w2<-log(x04w2+1)
#x05w2  local2 clocent  normal



##Generating categories in citations and IV that have power law (generate dummy variables
based on their distributions)
y1 <- ifelse(y>1, c(1), c(0))
```

```
x044<-ifelse(x04>2, c(1),c(0))
xx044<-ifelse(xx04>1, c(1),c(0))
x055<-ifelse(x05>4, c(1),c(0))
x066<-ifelse(x06>2, c(1),c(0))
x099<-ifelse(x09>0.1, c(1),c(0))
x011<-ifelse(x11>0.5, c(1),c(0))
x07w11<-log(x07w1)
x04w22<-ifelse(x04w2>1, c(1),c(0))
x07w22<-log(x07w2)

#Binding data into different Models

Xm<-cbind(x02, x044, x055, x07, x099, x10, xx044, xx09)
Xmm<-cbind(x02, x044, x055, x07, x099, x10)
Xw1<-cbind(x02w1, x03w1, x04w1, x05w1, x07w11, x09w1, x10w1, x11w1)
Xw11<-cbind(x03w1, x07w11)
Xw2<-cbind(x02w2, x03w2, x04w22, x05w2, x07w22, x09w2, x10w2, x11w2)
Xw22<-cbind(x02w2, x03w2, x04w22, x10w2)
Xc<-cbind(Xm,Xw1,Xw2)
Xcc<-cbind(Xmm,Xw11,Xw22)
Xcc1<-cbind(Xmm, Xw11)
Xc12<-cbind(Xw1, Xw2)
Xc12a<-cbind(Xw11,Xw22)
#get the rank of a matrix
rank<-rankMatrix(Xc, tol = NULL, method = c("tolNorm2", "qr.R", "qrLINPACK", "qr",
                                            "useGrad", "maybeGrad"),sval = svd(Xc, 0,
0)$d, warn.t = TRUE)  # 3
rank1<-as.data.frame(rank)
rank1
corr<-rcorr(as.matrix(Xc))
corr
rank2<-rankMatrix(Xcc, tol = NULL, method = c("tolNorm2", "qr.R", "qrLINPACK", "qr",
                                            "useGrad", "maybeGrad"),sval = svd(Xcc, 0,
0)$d, warn.t = TRUE)  # 3
rank3<-as.data.frame(rank2)
rank3
corr2<-rcorr(as.matrix(Xcc))
corr2
#
#Write variables in CSV file
write.csv(Xc, file="C:\\Users\\Philipmunyua\\Google Drive\\nanoHUB research\\02
analysis\\04 develop level _test spatial\\02 input data\\SEM.csv")
write.csv(Xcc, file="C:\\Users\\Philipmunyua\\Google Drive\\nanoHUB research\\02
analysis\\04 develop level _test spatial\\02 input data\\SEM1.csv")

#---Different Models as Matrix

Xc=as.matrix(Xc)
Xcc=as.matrix(Xcc)
Xc12=as.matrix(Xc12)
Xc12a=as.matrix(Xc12a)

Xcc1=as.matrix(Xcc1)
y1=as.matrix(y1)
dat = data.frame(y1,Xc)
dat1 = data.frame(y1,Xcc)
dat2 = data.frame(y1,Xc12)
dat3 = data.frame(y1,Xc12a)

#*Probit_Centralities_for_both_weights
probitw1w2<-glm(y1 ~Xc12)
summary(probitw1w2)
probitmfx(probitw1w2, data=dat1)

#**SEM Model
semprobit.fit1 <- semprobit(y1~Xc, gmat, ndraw=500, burn.in=100, thinning=1, prior=NULL)
summary(semprobit.fit1)

class(gmat)
```

```
semprobit.fit2 <- semprobit(y1~Xcc, gmat2, ndraw=1000, burn.in=100, thinning=1,
prior=NULL)
summary(semprobit.fit2)
logLik(semprobit.fit2)
AIC(semprobit.fit2)
marginal.effects(semprobit.fit2)
impacts.semprobit(semprobit.fit2)
yfit<-fitted.values(semprobit.fit2)
yfit<-as.matrix(yfit)
res<-y1-yfit
res1<-as.numeric(res)
class(res)
moran.test(res,nb1)
moran.plot(res1,nb1)

semprobit.fit.12a <- semprobit(y1~Xc12a, gmat2, ndraw=1000, burn.in=100, thinning=1,
prior=NULL)
summary(semprobit.fit.12a)
logLik(semprobit.fit.12a)
AIC(semprobit.fit.12a)

#marginal.effects(semprobit.fit2)
semprobit.fit2a <- semprobit(y1~Xcc1, gmat2, ndraw=1000, burn.in=100, thinning=1,
prior=NULL)
summary(semprobit.fit2a)
logLik(semprobit.fit2a)
AIC(semprobit.fit2a)

sem.probit3<-sem_probit_mcmc(y1, Xcc, gmat2, ndraw = 1000, burn.in = 100, thinning = 1,
                             prior=list(a1=1, a2=1, c=rep(0, ncol(X)),
T=diag(ncol(X))*1e12,
                                        nu = 0, d0 = 0, lflag = 0),
                             start = list(rho = 0.75, beta = rep(0, ncol(X)), sige = 1),
                             m=10, showProgress=FALSE)
summary(sem.probit3)

#LR
LR.test.stat.sem <- as.numeric(2*(logLik(semprobit.fit2) - logLik(semprobit.fit2a))) ##
getting the LR tests
print(LR.test.stat.sem)## display the LR tests
pchisq(LR.test.stat.sem, 1, lower=F) ## getting the P-value

##---------Interation Models


#Generating interaction terms with X077 (Citation dummy)
x703<-x077*x03 #Boncent and citation dummy
x705<-x077*x055 #Betcent and citation dummy
x706<-x077*x066 #clocent and citation dummy
x708<-x077*x08 # degree and citation dummy
x710<-x077*x100 # Eigenvector and citation dummy
x712<-x077*x112 #googlepgrank and citation dummy


#
X=cbind(x01,x022,x03,x04,x055,x066,x077,x08,x09,x100, x11, x112, x703, x705, x706, x708,
x710, x712)
Model_1a=cbind(x03,x04,x055, x066, x077, x08, x100, x112)
Model_1b=cbind(x03,x04,x055, x066, x08, x100, x112)
Model_1c=cbind(x03,x04,x055, x066, x077, x08, x100)
Model_1d=cbind(x03,x04,x055, x066, x077, x08, x100, x112, x703, x705, x706, x708, x710,
x712)
Model_1s=cbind(x04,x055, x066, x077, x08, x100)
Model_1si=cbind(x04,x055, x066, x077, x08, x100, x705, x706, x708, x710)
Model_1sr=cbind(x04,x055, x066, x08, x100)

X=as.matrix(X)
y1=as.matrix(y1)
dat = data.frame(y1,X)
```

```
##Save data to stata format
library(foreign)
write.dta (dat, "C:/Users/Philipmunyua/Google Drive/nanoHUB research/02 analysis/04
develop level _test spatial/02 input data/dat.dta")
##End save data to stata format

#Probit for model 1c
probit1a<-glm(y1 ~Model_1a)
summary(probit1a)
probitmfx(probit1a, data=dat)

#mixed model
#probit1e<-lmer(y1 ~Model 1a + (1|x703) + (1|x705), data=dat)
#summary(probit1e)

##Spatial Auto Regressive (SAR) probit bayesian-based on social networksarprobit.fit1a <-
sarprobit(y1~ Model_1a, gmat, ndraw =1000, burn.in = 477,thinning = 1)

sarprobit.fit1a <- sarprobit(y1~ Model_1a, gmat, ndraw =1000, burn.in = 477,thinning = 1)
summary(sarprobit.fit1a)
logLik(sarprobit.fit1a)
AIC(sarprobit.fit1a)
marginal.effects(sarprobit.fit1a)

#Probit for model 1b
probit1b<-glm(y1 ~Model_1b)
summary(probit1b)
probitmfx(probit1b, data=dat)

#**
sarprobit.fit1b <- sarprobit(y1~ Model_1b, gmat, ndraw =1000, burn.in = 477,thinning = 1)
summary(sarprobit.fit1b)
logLik(sarprobit.fit1b)
AIC(sarprobit.fit1b)
marginal.effects(sarprobit.fit1b)

#Probit for model 1c
probit1c<-glm(y1 ~Model_1c)
summary(probit1c)
probitmfx(probit1c, data=dat)

#**
sarprobit.fit1c <- sarprobit(y1~ Model_1c, gmat, ndraw =1000, burn.in = 477,thinning = 1)
summary(sarprobit.fit1c)
logLik(sarprobit.fit1c)
AIC(sarprobit.fit1c)
marginal.effects(sarprobit.fit1c)

#Probit for model 1d
probit1d<-glm(y1 ~Model_1d)
summary(probit1d)
probitmfx(probit1d, data=dat)

#**
sarprobit.fit1d <- sarprobit(y1~ Model_1d, gmat, ndraw =1000, burn.in = 477,thinning = 1)
summary(sarprobit.fit1d)
logLik(sarprobit.fit1d)
AIC(sarprobit.fit1d)
marginal.effects(sarprobit.fit1d)

#Probit for model 1S
probit1s<-glm(y1 ~Model_1s)
summary(probit1s)
probitmfx(probit1s, data=dat)


#**
sarprobit.fit1s <- sarprobit(y1~ Model_1s, gmat, ndraw =1000, burn.in = 477,thinning = 1)
```

```
summary(sarprobit.fit1s)
logLik(sarprobit.fit1s)
AIC(sarprobit.fit1s)
marginal.effects(sarprobit.fit1s)

#Probit for model 1Sr
probit1sr<-glm(y1 ~Model_1sr)
summary(probit1sr)
probitmfx(probit1sr, data=dat)

#**
sarprobit.fit1sr <- sarprobit(y1~ Model_1sr, gmat, ndraw =1000, burn.in = 477,thinning =
1)
summary(sarprobit.fit1sr)
logLik(sarprobit.fit1sr)
AIC(sarprobit.fit1sr)

#Probit for model 1si
probit1si<-glm(y1 ~Model_1si)
summary(probit1si)
probitmfx(probit1si, data=dat)

#**
sarprobit.fit1si <- sarprobit(y1~ Model_1si, gmat, ndraw =1000, burn.in = 477,thinning =
1)
summary(sarprobit.fit1si)
logLik(sarprobit.fit1si)
AIC(sarprobit.fit1si)
marginal.effects(sarprobit.fit1si)


##Likelihood ratio test
LR.test.stat.s <- as.numeric(2*(logLik(sarprobit.fit1s) - logLik(sarprobit.fit1sr))) ##
getting the LR tests
print(LR.test.stat.s)## display the LR tests
pchisq(LR.test.stat.s, 1, lower=F) ## getting the P-value


##Likelihood ratio test based on authorship dummy restriction
LR.test.stat2 <- as.numeric(2*(logLik(probit1s) - logLik(probit1sr))) ##  getting the LR
tests
print(LR.test.stat2)## display the LR tests
pchisq(LR.test.stat2, 1, lower=F) ## getting the P-value

##Likelihood ratio test of interaction term model
LR.test.stat3 <- as.numeric(2*(logLik(probit1si) - logLik(probit1s))) ##  getting the LR
tests
print(LR.test.stat3)## display the LR tests
pchisq(LR.test.stat3, 1, lower=F) ## getting the P-value
```

##Essay

```
## Least Square Iterative Process for estimating degree of attachment (randomeness to
preferential)
#--load data set (data frame)

deg06<-read.csv("t06.csv",header=TRUE)

# converting data in table format and generate frequency  distribution
mytable <- table(deg06$In)
relFreq <- prop.table(mytable) #Generating frequency distribution
dd<-as.data.frame(relFreq) #converting data to data frame
dd<- rename(dd, c(Var1="degree"))

write.csv(dd, file="C:\\Users\\Philipmunyua\\Google Drive\\nanoHUB research\\02
analysis\\04 develop level _test spatial\\02 input data\\t06_deg_freq.csv")
dat<-read.csv("t06_deg_freq.csv")
```

```
dat$X<-NULL
dat$ln_deg<-log(dat$degree)
dat$ln_freq<-log(dat$Freq)

# Generating alpha and model distribution
summary(dat)
a<-0.98
a1<-range(0.000,1.000)
m<-(weighted.mean(dat$degree, dat$Freq)*0.5)
d<-as.matrix(dat$degree)
fd<-as.matrix(dat$Freq)
xd<-as.matrix(dat$degree+((2*m*a)/(1-a)))
y<-log(fd)
x<-log(xd)
fit <- lm(y ~ x, data=dat)
#summary(fit) # show results
b.fit<-coef(fit)
b<-b.fit[2]
t<-((b-2)/b)
a1<-ifelse(t>max(a1),max(a1),t)

#model distribution
#Random model F(d)=1-(e-(d-m/m))
tau<--((d-m)/m)
fd2<-1-exp(tau)
dat$fd2<-fd2
dat$ln_fd2<-log(dat$fd2)
dat$ln_fd2[is.nan(dat$ln_fd2)] <--4 ##replacing NaNs with value
write.csv(dat, file="C:\\Users\\Philipmunyua\\Google Drive\\nanoHUB research\\02
analysis\\04 develop level _test spatial\\02 input data\\t06_dat.csv")


##-------------------ERGM Model


### Converting from Matrix to ListW object for spatial regression ###

## set the working directory to Nanohub
rm(list=ls())
setwd("C:\\Users\\mutuma15\\Google Drive\\nanoHUB research\\02 analysis\\04 develop level
_test spatial\\02 input data")
setwd("C:\\Users\\pmunyua\\Google Drive\\nanoHUB research\\02 analysis\\04 develop level
_test spatial\\02 input data")
setwd("C:\\Users\\Philipmunyua\\Google Drive\\nanoHUB research\\02 analysis\\04 develop
level _test spatial\\02 input data")

library(spdep)
library(Matrix)
library(igraph)
library(lmtest)
library(sphet)
library(AER)
library(spatstat)
library(spatialprobit)
library(McSpatial)
library(mfx)
library(stats)
library(Hmisc)
library(utils)
library(Zelig)
library(base)
library(reshape)
library(ggplot2)
library(statnet)
library(network)
library(ergm)
library(sna)
library(coda)
```

```
graph_list<-read.csv("time_slice_2.csv") ## logged edge_list
graph_list$X<-NULL
graph_list$WEIGHT<-log(1+graph_list$WEIGHT)#logging the weights
mat<-as.matrix(graph_list[,1:2])

## Make an directed graph
graph<-graph.edgelist(mat,directed=TRUE)
E(graph)$weight<-graph_list[,3]
gmat<-get.adjacency(graph,attr="weight")
class(graph)
class(gmat)
adjmat<-as.matrix(gmat)
gmat1<-network(gmat)

###-------KS test for Stochastic Dominance-------------------------#

## set the working directory to Nanohub
setwd("C:\\Users\\Philipmunyua\\Google Drive\\nanoHUB research\\02 analysis\\04 develop
level _test spatial\\02 input data")

#load data

dat<-read.csv("distributions2_8.csv")
gp2<-dat$freq2
gp3<-dat$freq3
gp4<-dat$freq4
gp5<-dat$freq5
gp6<-dat$freq6
gp7<-dat$freq7
gp8<-dat$freq8


#KS-Test
ks23<-ks.test(gp2, gp3, alternative="two.sided")
ks24<-ks.test(gp2, gp4, alternative="two.sided")
ks25<-ks.test(gp2, gp5, alternative="two.sided")
ks26<-ks.test(gp2, gp6, alternative="two.sided")
ks27<-ks.test(gp2, gp7, alternative="two.sided")
ks28<-ks.test(gp2, gp8, alternative="two.sided")
#3
ks34<-ks.test(gp3, gp4, alternative="two.sided")
ks35<-ks.test(gp3, gp5, alternative="two.sided")
ks36<-ks.test(gp3, gp6, alternative="two.sided")
ks37<-ks.test(gp3, gp7, alternative="two.sided")
ks38<-ks.test(gp3, gp8, alternative="two.sided")
#4
ks45<-ks.test(gp4, gp5, alternative="two.sided")
ks46<-ks.test(gp4, gp6, alternative="two.sided")
ks47<-ks.test(gp4, gp7, alternative="two.sided")
ks48<-ks.test(gp4, gp8, alternative="two.sided")
#5
ks56<-ks.test(gp5, gp6, alternative="two.sided")
ks57<-ks.test(gp5, gp7, alternative="two.sided")
ks58<-ks.test(gp5, gp8, alternative="two.sided")
#6
ks67<-ks.test(gp6, gp7, alternative="two.sided")
ks68<-ks.test(gp6, gp8, alternative="two.sided")
#7
ks78<-ks.test(gp7, gp8, alternative="two.sided")


ks23
ks24
ks25
ks26
ks27
ks28
```

```
ks34
ks35
ks36
ks37
ks38

ks45
ks46
ks47
ks48

ks56
ks57
ks58

ks67
ks68

ks78

#---------Dominating Distribution Statistics


### Converting from Matrix to ListW object for spatial regression ###

## set the working directory to Nanohub
rm(list=ls())
setwd("C:\\Users\\mutuma15\\Google Drive\\nanoHUB research\\02 analysis\\04 develop level
_test spatial\\02 input data")
setwd("C:\\Users\\pmunyua\\Google Drive\\nanoHUB research\\02 analysis\\04 develop level
_test spatial\\02 input data")
setwd("C:\\Users\\Philipmunyua\\Google Drive\\nanoHUB research\\02 analysis\\04 develop
level _test spatial\\02 input data")

library(spdep)
library(Matrix)
library(igraph)
library(lmtest)
library(sphet)
library(AER)
library(spatstat)
library(spatialprobit)
library(McSpatial)
library(mfx)
library(stats)
library(Hmisc)
library(utils)
library(Zelig)
library(base)
library(reshape)
library(ggplot2)
library(boot)

graph_list<-read.csv("time_slice_2.csv") ## logged edge_list
graph_list$X<-NULL
graph_list$WEIGHT<-log(1+graph_list$WEIGHT)#logging the weights
mat<-as.matrix(graph_list[,1:2])

## Make an directed graph
graph<-graph.edgelist(mat,directed=TRUE)
E(graph)$weight<-graph_list[,3]
gmat<-get.adjacency(graph,attr="weight")
class(graph)
class(gmat)
adjmat<-as.matrix(gmat)

for(i in 1:477)
{
  adjmat[i,i]= 1.0
```

```
}

nb<-mat2listw(adjmat)
class(nb)
## Row standardized weigth matrix:Row standardization creates proportional
##weights in cases where features have an unequal number of neighbors
nb1<-nb2listw(nb$neighbours, style="W")
class(nb1)


#Degree Centrality (p. 69)
deg<-degree(gmat, v=V(graph),
mode=c("all","out","in","total"),loops=TRUE,normalized=FALSE)
deg1<-as.data.frame(deg)
deg.dis<-degree.distribution(graph, cumulative=FALSE)
dd<-as.data.frame(deg.dis)
dd$degree<-c(1:339)
dd$Freq<-dd[,1]
dd$ln_degree<-log(dd$degree)
dd$ln_Freq<-log(dd$Freq)
dd$deg.dis<-NULL

write.csv(dd, file="C:\\Users\\Philipmunyua\\Google Drive\\nanoHUB research\\02
analysis\\04 develop level _test spatial\\02 input data\\ts_02_dat.csv")
dat<-read.csv("ts_02_dat.csv")
dat$X<-NULL

# Generating alpha and model distribution
summary(dat)
dat1 <- dat[which(dat$Freq>0.0000000),]
sum(dat1$Freq)
summary(dat1)

a<-0.998
a1<-range(0.000,1.000)
m<-(weighted.mean(dat1$degree, dat1$Freq)*0.5)
d<-as.matrix(dat1$degree)
fd<-as.matrix(dat1$Freq)
xd<-as.matrix(dat1$degree+((2*m*a)/(1-a)))
y<-log(fd)
x<-log(xd)
fit <- lm(y ~ x, data=dat1)
#summary(fit) # show results
b.fit<-coef(fit)
b<-b.fit[2]
t<-((b-2)/b)
a1<-ifelse(t>max(a1),max(a1),t)

#model distribution
#Hybrid model F(d)=1-((m+(2am/1-a))/(d+(2am/1-a)))^2/(1-a)
a<-0.999
tau<-(2/(1-a))
tau1<-(2*a*m)/(1-a)
num<-m+tau1
den<-d+tau1
fd2<-range(0, 1000)
con<-1-(num/den)^tau
fd2<-ifelse(con>=min(fd2),con,min(fd2))
dat1$fd2<-fd2
dat1$ln_fd2<-log(dat1$fd2)
dat1$ln_fd2[dat1$ln_fd2==-Inf] <-dat1$ln_Freq###replacing infinit with value
corr<-rcorr(dat1$ln_Freq, dat1$ln_fd2)
corr
dat1$ff<-dat1$ln_fd2
dat1$ff[dat1$ff==-2.9] <-dat1$ln_Freq##replacing NaNs with value

write.csv(dat1, file="C:\\Users\\Philipmunyua\\Google Drive\\nanoHUB research\\02
analysis\\04 develop level _test spatial\\02 input data\\ts_02_dat1.csv")
```

```
attach(dat1)
plot(ln_degree, ln_Freq, main="Time slice02", xlab="Log(degree)", ylab="log(Frequency) ",
pch=19)#,the data plot
summary(dat1)
class(dat1)

y1<-as.matrix(dat1$ln_Freq)
x1<-as.matrix(dat1$ln_degree)
y<-as.matrix(dat1$Freq)
d<-as.matrix(dat1$degree)
cr<-fit <- lm(x1 ~ y1, data=dat1)
summary(cr)

#-------------Fitting results to models via bootstrap
#---Bootstrap (ln_degree)
n <-length(dat1$ln_degree)
B <- 1000
results <- rep(NA, B)
for (i in 1:B){
  boot.sample <- sample(n, replace=TRUE)
  results[i] <- mean(dat1$ln_degree[boot.sample])
}
with(dat1, mean(ln_degree) + c(-1,1)*2*sd(results))

bb<-mean(results)
bb
#---Bootstraping the variance

var.boot <- function(x,i){var(y1[i])}
boot<-boot(dat1,var.boot,1000)
out <- boot(dat1,var.boot,1000)
out
ci.var<-boot.ci(out,type="perc")
ci.var
hist(out$t)
hist(out$t, xlim=c(0.42,1.3), nclass=30, col=3, main="Histogram of Randomly Generated
Data for Variance")
abline(v=q95.np,lty=2)
abline(v=c(ci.var))
abline(v=c(ci.u,ci.l))
hist(theta.rand.median, xlim=c(-.2,.2), nclass=100, col=3, main="Histogram of Randomly
Generated Data for Medians")

hist(theta.rand.mean, xlim=c(-.2,.2), nclass=50, col=3, main="Histogram of Randomly
Generated Data for Means")
abline(v=c(ci.u,ci.l))
#---Bootstraping the Median
var.boot.1 <- function(x,i){median(y1[i])}
boot.1<-boot(dat1,var.boot.1,1000)
boot.1
out.1 <- boot(dat1,var.boot.1,1000)
ci.var.1<-boot.ci(out.1,type="perc")
ci.var.1
hist(out.1$t)
#---Bootstraping the Mean
var.boot.2 <- function(x,i){mean(y1[i])}
boot.2<-boot(dat1,var.boot.2, 1000)
boot.2
out.2 <- boot(dat1,var.boot.2, 1000)
ci.var.2<-boot.ci(out.2,type="perc")
ci.var.2
hist(out.2$t)
#--Bootstrapping the Sample Median--1

ns<-1000
res<-numeric(ns)
for (i in 1:ns) {
res[i] <- median(sample(y1, replace=T))
se.b<-sqrt(var(res))
```

```
}
se.b
quantile(res, p = c(0.025, 0.975))
par(mfrow=c(1,1))
hist(res)
qqnorm(res)

#-Bootstrapping a Trimmed Mean

tm <- mean(y1, trim = 0.10)
nsamp <- 1000
res <- numeric(nsamp)
for (i in 1:nsamp) {
  res[i] <- mean(sample(y1, replace = TRUE), trim=0.10)
}
hist(res)
abline(v = tm, lty = 4)
sd(res)
quantile(res, p = c(0.05, 0.95))


#-------Bootstrap degree
#---Bootstraping the variance

var.boot <- function(x,i){var(d[i])}
boot<-boot(dat1,var.boot,1000)
out <- boot(dat1,var.boot,1000)
out
ci.var<-boot.ci(out,type="perc")
ci.var
hist(out$t)

#---Bootstraping the Median
var.boot.1 <- function(x,i){median(d[i])}
boot.1<-boot(dat1,var.boot.1,1000)
boot.1
out.1 <- boot(dat1,var.boot.1,1000)
ci.var.1<-boot.ci(out.1,type="perc")
ci.var.1
hist(out.1$t)
#---Bootstraping the Mean
var.boot.2 <- function(x,i){mean(d[i])}
boot.2<-boot(dat1,var.boot.2, 1000)
boot.2
out.2 <- boot(dat1,var.boot.2, 1000)
ci.var.2<-boot.ci(out.2,type="perc")
ci.var.2
hist(out.2$t)

#----End

dat2 <- dat1[,c("ln_degree", "ln_Freq", "ff")]
x1<-as.matrix(dat2$ln_degree)
y1<-as.matrix(dat2$ln_Freq)
x2<-as.matrix(dat2$ln_degree)
y2<-as.matrix(dat2$ff)
class(x1)

plot(x1,y1,xlim=range(c(x1,x2)),ylim=range(c(y1,y2)),col="red")
points(x2,y2,col="blue")

#compling vector of new variables
object<-data.frame(BC,BeC,CC, dC,evcent1, dcomp, googpr.old1)
file_cc<-file("C:\\Users\\Philipmunyua\\Google Drive\\nanoHUB research\\02 analysis\\04
develop level _test spatial\\02 input data\\centrality1219.csv","w")
write.csv(object, file_cc)
close(file_cc)

#Merging data sets
```

```
object1<-read.csv("centrality1219.csv",header=TRUE)
colnames(object1)[1] <- "Dev_Name" #Renaming a column
class(object1)
newdev<-merge(dev, object1, by="Dev_Name") #Merging developer network with the new
variables by the common variable "Dev_Name"
newdev$X<-NULL
class(newdev)

file_dev<-file("C:\\Users\\Philipmunyua\\Google Drive\\nanoHUB research\\02 analysis\\04
develop level _test spatial\\02 input data\\dev_attrib_ver1219.csv","w")
write.csv(newdev, file_dev)
close(file_dev)

##Test for powerlaw for indegree (d=degree and x1=log indegree)
#Fitting Power Law (p.239)
#fitting a power-law distribution
#d-degree
powerlawfit<-power.law.fit(d, dmin=NULL, start=2, force.continuous=FALSE,
implementation=c("plfit", "R.mle"))
powerlawx<-as.data.frame(powerlawfit)
#x1-log_indegree
powerlawfit1<-power.law.fit(x1, x1min=NULL, start=2, force.continuous=FALSE,
implementation=c("plfit", "R.mle"))
powerlawx1<-as.data.frame(powerlawfit)

###Timeline descriptives
#Assortivity
V(graph)$foo <- sample(1:3, replace=TRUE, vcount(graph))
assort<-assortativity.nominal(graph, types=V(graph)$foo)

#clusters
#calculates the maximal (weakly or strongly) connected components of a graph
isclus<-is.connected(graph, mode=c("weak", "strong")) #decided whether the graph is
weakly or strongly connected

#Diameter
#calculates the length of the longest geodesic
getdiam<-get.diameter(graph, directed=TRUE, unconnected=TRUE, weights=NULL) # retunrs a
path with actual diameter
getdiam1<-as.data.frame(getdiam)

#Dyad Census
dyads<-dyad.census(graph)
dyads1<-as.data.frame(dyads)

#Graph density
#Density is the ratio of the number of edges (links) and the number of possible edges
density<-graph.density(graph, loops=FALSE)
density1<-as.data.frame(density)

#Average nearest neighbor degree
# calculates  the average nearest neighbor degree of the given vertices and the same
quantity in the function of the vertex degree
avneigh<-graph.knn(graph, vids=V(graph),weights=NULL)
avneigh1<-as.data.frame(avneigh)

#Neighborhood of graph vertices
#finds nodes that are not farther than a given limit from another fixed node
(neighborhood of the node)
neigh.size<-neighborhood.size(graph, 1, nodes=V(graph), mode=c("all","out","in")) #
calculates the size of neighborhood
neigh.size1<-as.data.frame(neigh.size)

#Reciprocity of graphs
recipro<-reciprocity(graph, ignore.loops=TRUE, mode=c("default", "ratio"))
recipro1<-as.data.frame(recipro)

#Shortest Path
#Calculates the shortest paths between vertices
```

```
shortpath<-shortest.paths(graph, v=V(graph),mode=c("all","out","in"),
                          weights=NULL,
algorithm=c("automatic","unweighted","dijkstra","bellman-ford","johnson"))
getshortpath<-get.shortest.paths(graph, 2, to=V(graph), mode = c("out", "all",
                                                                 "in"), weights = NULL,
output=c("vpath", "epath", "both"), predecessors = FALSE, inbound.edges = FALSE)
getallshortpath<-get.all.shortest.paths(graph, 2, to = V(graph), mode = c("out",
                                                                          "all", "in"),
weights=NULL)
avshortpath<-average.path.length(graph, directed=TRUE, unconnected=TRUE)
avshortpath1<-as.data.frame(avshortpath)
pathlenngthhist<-path.length.hist (graph, directed = TRUE)

#Transivity or clustering coefficient
#A measure of the probability that the adjacency nodes of a node are connected (also
known as clustering coefficient)
clustcoeff<-transitivity(graph, type=c("undirected", "global", "globalundirected",
                                       "localundirected", "local", "average",
"localaverage",
                                       "localaverageundirected", "barrat", "weighted"),
vids=NULL,
                         weights=NULL, isolates=c("NaN", "zero"))
clustcoeff1<-as.data.frame(clustcoeff)

#Triad Census
triads<-triad.census(graph)


##----------ERGM Models

ERGM1<-gmat1~mutual
ERGM2<-gmat1~mutual+transitive
ERGM3<-gmat1~mutual+istar(3)+transitive
ERGM4<-gmat1~mutual+gwidegree(2.5, fixed=TRUE)
ERGM5<-gmat1~edges+mutual
ERGM6<-gmat1~edges+mutual+transitive
ERGM7<-gmat1~edges+mutual+istar(3)+transitive
ERGM8<-gmat1~edges+mutual+gwidegree(2.5, fixed=TRUE)

ERGM.Model.1<-ergm(ERGM1)
summary(ERGM.Model.1)
mcmc.diagnostics(ERGM.Model.1)
gof.1<-gof(ERGM.Model.1)
summary(gof.1)
plot(gof.1)
ERGM.Model.2<-ergm(ERGM2)
summary(ERGM.Model.2)
gof.2<-gof(ERGM.Model.2)
summary(gof.2)
plot(gof.2)

ERGM.Model.3<-ergm(ERGM3)
summary(ERGM.Model.3)
plot(ERGM.Model.3$sample, ask=FALSE)
gof.3<-gof(ERGM.Model.3)
plot(gof.3)
summary(gof.3)
ERGM.Model.4<-ergm(ERGM4)
summary(ERGM.Model.4)
gof.4<-gof(ERGM.Model.4)
plot(gof.4)
summary(gof.4)
mcmc.diagnostics(ERGM.Model.4)
ERGM.Model.5<-ergm(ERGM5)
summary(ERGM.Model.5)
gof.5<-gof(ERGM.Model.5)
plot(gof.5)
summary(gof.5)
mcmc.diagnostics(ERGM.Model.5)
```

```
ERGM.Model.6<-ergm(ERGM6)
summary(ERGM.Model.6)
gof.6<-gof(ERGM.Model.6)
plot(gof.6)
summary(gof.6)
mcmc.diagnostics(ERGM.Model.6)

ERGM.Model.7<-ergm(ERGM7)
summary(ERGM.Model.7)
gof.7<-gof(ERGM.Model.7)
plot(gof.7)
summary(gof.7)
mcmc.diagnostics(ERGM.Model.7)

ERGM.Model.8<-ergm(ERGM8)
summary(ERGM.Model.8)
gof.8<-gof(ERGM.Model.8)
plot(gof.8)
summary(gof.8)
mcmc.diagnostics(ERGM.Model.8)
#Anova test for models
Anova12<-anova(ERGM.Model.1,ERGM.Model.2)
Anova12
Anova13<-anova(ERGM.Model.1,ERGM.Model.3)
Anova13
Anova14<-anova(ERGM.Model.1,ERGM.Model.4)
Anova14

###Essay #3
##Essay

## Least Square Iterative Process for estimating degree of attachment (randomeness to
preferential)
#--load data set (data frame)

deg06<-read.csv("t06.csv",header=TRUE)

# converting data in table format and generate frequency  distribution
mytable <- table(deg06$In)
relFreq <- prop.table(mytable) #Generating frequency distribution
dd<-as.data.frame(relFreq) #converting data to data frame
dd<- rename(dd, c(Var1="degree"))

write.csv(dd, file="C:\\Users\\Philipmunyua\\Google Drive\\nanoHUB research\\02
analysis\\04 develop level _test spatial\\02 input data\\t06_deg_freq.csv")
dat<-read.csv("t06_deg_freq.csv")
dat$X<-NULL
dat$ln_deg<-log(dat$degree)
dat$ln_freq<-log(dat$Freq)

# Generating alpha and model distribution
summary(dat)
a<-0.98
a1<-range(0.000,1.000)
m<-(weighted.mean(dat$degree, dat$Freq)*0.5)
d<-as.matrix(dat$degree)
fd<-as.matrix(dat$Freq)
xd<-as.matrix(dat$degree+((2*m*a)/(1-a)))
y<-log(fd)
x<-log(xd)
fit <- lm(y ~ x, data=dat)
#summary(fit) # show results
b.fit<-coef(fit)
b<-b.fit[2]
t<-((b-2)/b)
a1<-ifelse(t>max(a1),max(a1),t)

#model distribution
```

```
#Random model F(d)=1-(e-(d-m/m))
tau<--((d-m)/m)
fd2<-1-exp(tau)
dat$fd2<-fd2
dat$ln_fd2<-log(dat$fd2)
dat$ln_fd2[is.nan(dat$ln_fd2)] <--4 ##replacing NaNs with value
write.csv(dat, file="C:\\Users\\Philipmunyua\\Google Drive\\nanoHUB research\\02
analysis\\04 develop level _test spatial\\02 input data\\t06_dat.csv")


##-------------------ERGM Model


### Converting from Matrix to ListW object for spatial regression ###

## set the working directory to Nanohub
rm(list=ls())
setwd("C:\\Users\\mutuma15\\Google Drive\\nanoHUB research\\02 analysis\\04 develop level
_test spatial\\02 input data")
setwd("C:\\Users\\pmunyua\\Google Drive\\nanoHUB research\\02 analysis\\04 develop level
_test spatial\\02 input data")
setwd("C:\\Users\\Philipmunyua\\Google Drive\\nanoHUB research\\02 analysis\\04 develop
level _test spatial\\02 input data")

library(spdep)
library(Matrix)
library(igraph)
library(lmtest)
library(sphet)
library(AER)
library(spatstat)
library(spatialprobit)
library(McSpatial)
library(mfx)
library(stats)
library(Hmisc)
library(utils)
library(Zelig)
library(base)
library(reshape)
library(ggplot2)
library(statnet)
library(network)
library(ergm)
library(sna)
library(coda)


graph_list<-read.csv("time_slice_2.csv") ## logged edge_list
graph_list$X<-NULL
graph_list$WEIGHT<-log(1+graph_list$WEIGHT)#logging the weights
mat<-as.matrix(graph_list[,1:2])

## Make an directed graph
graph<-graph.edgelist(mat,directed=TRUE)
E(graph)$weight<-graph_list[,3]
gmat<-get.adjacency(graph,attr="weight")
class(graph)
class(gmat)
adjmat<-as.matrix(gmat)
gmat1<-network(gmat)

###-------KS test for Stochastic Dominance-------------------------#

## set the working directory to Nanohub
setwd("C:\\Users\\Philipmunyua\\Google Drive\\nanoHUB research\\02 analysis\\04 develop
level _test spatial\\02 input data")

#load data
```

```
dat<-read.csv("distributions2_8.csv")
gp2<-dat$freq2
gp3<-dat$freq3
gp4<-dat$freq4
gp5<-dat$freq5
gp6<-dat$freq6
gp7<-dat$freq7
gp8<-dat$freq8


#KS-Test
ks23<-ks.test(gp2, gp3, alternative="two.sided")
ks24<-ks.test(gp2, gp4, alternative="two.sided")
ks25<-ks.test(gp2, gp5, alternative="two.sided")
ks26<-ks.test(gp2, gp6, alternative="two.sided")
ks27<-ks.test(gp2, gp7, alternative="two.sided")
ks28<-ks.test(gp2, gp8, alternative="two.sided")
#3
ks34<-ks.test(gp3, gp4, alternative="two.sided")
ks35<-ks.test(gp3, gp5, alternative="two.sided")
ks36<-ks.test(gp3, gp6, alternative="two.sided")
ks37<-ks.test(gp3, gp7, alternative="two.sided")
ks38<-ks.test(gp3, gp8, alternative="two.sided")
#4
ks45<-ks.test(gp4, gp5, alternative="two.sided")
ks46<-ks.test(gp4, gp6, alternative="two.sided")
ks47<-ks.test(gp4, gp7, alternative="two.sided")
ks48<-ks.test(gp4, gp8, alternative="two.sided")
#5
ks56<-ks.test(gp5, gp6, alternative="two.sided")
ks57<-ks.test(gp5, gp7, alternative="two.sided")
ks58<-ks.test(gp5, gp8, alternative="two.sided")
#6
ks67<-ks.test(gp6, gp7, alternative="two.sided")
ks68<-ks.test(gp6, gp8, alternative="two.sided")
#7
ks78<-ks.test(gp7, gp8, alternative="two.sided")


ks23
ks24
ks25
ks26
ks27
ks28

ks34
ks35
ks36
ks37
ks38

ks45
ks46
ks47
ks48

ks56
ks57
ks58

ks67
ks68

ks78

#---------Dominating Distribution Statistics
```

```
### Converting from Matrix to ListW object for spatial regression ###

## set the working directory to Nanohub
rm(list=ls())
setwd("C:\\Users\\mutuma15\\Google Drive\\nanoHUB research\\02 analysis\\04 develop level
_test spatial\\02 input data")
setwd("C:\\Users\\pmunyua\\Google Drive\\nanoHUB research\\02 analysis\\04 develop level
_test spatial\\02 input data")
setwd("C:\\Users\\Philipmunyua\\Google Drive\\nanoHUB research\\02 analysis\\04 develop
level _test spatial\\02 input data")

library(spdep)
library(Matrix)
library(igraph)
library(lmtest)
library(sphet)
library(AER)
library(spatstat)
library(spatialprobit)
library(McSpatial)
library(mfx)
library(stats)
library(Hmisc)
library(utils)
library(Zelig)
library(base)
library(reshape)
library(ggplot2)
library(boot)

graph_list<-read.csv("time_slice_2.csv") ## logged edge_list
graph_list$X<-NULL
graph_list$WEIGHT<-log(1+graph_list$WEIGHT)#logging the weights
mat<-as.matrix(graph_list[,1:2])

## Make an directed graph
graph<-graph.edgelist(mat,directed=TRUE)
E(graph)$weight<-graph_list[,3]
gmat<-get.adjacency(graph,attr="weight")
class(graph)
class(gmat)
adjmat<-as.matrix(gmat)

for(i in 1:477)
{
  adjmat[i,i]= 1.0
}

nb<-mat2listw(adjmat)
class(nb)
## Row standardized weigth matrix:Row standardization creates proportional
##weights in cases where features have an unequal number of neighbors
nb1<-nb2listw(nb$neighbours, style="W")
class(nb1)


#Degree Centrality (p. 69)
deg<-degree(gmat, v=V(graph),
mode=c("all","out","in","total"),loops=TRUE,normalized=FALSE)
deg1<-as.data.frame(deg)
deg.dis<-degree.distribution(graph, cumulative=FALSE)
dd<-as.data.frame(deg.dis)
dd$degree<-c(1:339)
dd$Freq<-dd[,1]
dd$ln_degree<-log(dd$degree)
dd$ln_Freq<-log(dd$Freq)
dd$deg.dis<-NULL
```

```
write.csv(dd, file="C:\\Users\\Philipmunyua\\Google Drive\\nanoHUB research\\02
analysis\\04 develop level _test spatial\\02 input data\\ts_02_dat.csv")
dat<-read.csv("ts_02_dat.csv")
dat$X<-NULL

# Generating alpha and model distribution
summary(dat)
dat1 <- dat[which(dat$Freq>0.0000000),]
sum(dat1$Freq)
summary(dat1)

a<-0.998
a1<-range(0.000,1.000)
m<-(weighted.mean(dat1$degree, dat1$Freq)*0.5)
d<-as.matrix(dat1$degree)
fd<-as.matrix(dat1$Freq)
xd<-as.matrix(dat1$degree+((2*m*a)/(1-a)))
y<-log(fd)
x<-log(xd)
fit <- lm(y ~ x, data=dat1)
#summary(fit) # show results
b.fit<-coef(fit)
b<-b.fit[2]
t<-((b-2)/b)
a1<-ifelse(t>max(a1),max(a1),t)

#model distribution
#Hybrid model F(d)=1-((m+(2am/1-a))/(d+(2am/1-a)))^2/(1-a)
a<-0.999
tau<-(2/(1-a))
tau1<-(2*a*m)/(1-a)
num<-m+tau1
den<-d+tau1
fd2<-range(0, 1000)
con<-1-(num/den)^tau
fd2<-ifelse(con>=min(fd2),con,min(fd2))
dat1$fd2<-fd2
dat1$ln_fd2<-log(dat1$fd2)
dat1$ln_fd2[dat1$ln_fd2==-Inf] <-dat1$ln_Freq###replacing infinit with value
corr<-rcorr(dat1$ln_Freq, dat1$ln_fd2)
corr
dat1$ff<-dat1$ln_fd2
dat1$ff[dat1$ff==-2.9] <-dat1$ln_Freq##replacing NaNs with value

write.csv(dat1, file="C:\\Users\\Philipmunyua\\Google Drive\\nanoHUB research\\02
analysis\\04 develop level _test spatial\\02 input data\\ts_02_dat1.csv")

attach(dat1)
plot(ln_degree, ln_Freq, main="Time slice02", xlab="Log(degree)", ylab="log(Frequency) ",
pch=19)#,the data plot
summary(dat1)
class(dat1)

y1<-as.matrix(dat1$ln_Freq)
x1<-as.matrix(dat1$ln_degree)
y<-as.matrix(dat1$Freq)
d<-as.matrix(dat1$degree)
cr<-fit <- lm(x1 ~ y1, data=dat1)
summary(cr)

#-------------Fitting results to models via bootstrap
#---Bootstrap (ln degree)
n <-length(dat1$ln_degree)
B <- 1000
results <- rep(NA, B)
for (i in 1:B){
  boot.sample <- sample(n, replace=TRUE)
  results[i] <- mean(dat1$ln_degree[boot.sample])
}
```

```
with(dat1, mean(ln_degree) + c(-1,1)*2*sd(results))

bb<-mean(results)
bb
#---Bootstraping the variance

var.boot <- function(x,i){var(y1[i])}
boot<-boot(dat1,var.boot,1000)
out <- boot(dat1,var.boot,1000)
out
ci.var<-boot.ci(out,type="perc")
ci.var
hist(out$t)
hist(out$t, xlim=c(0.42,1.3), nclass=30, col=3, main="Histogram of Randomly Generated
Data for Variance")
abline(v=q95.np,lty=2)
abline(v=c(ci.var))
abline(v=c(ci.u,ci.l))
hist(theta.rand.median, xlim=c(-.2,.2), nclass=100, col=3, main="Histogram of Randomly
Generated Data for Medians")

hist(theta.rand.mean, xlim=c(-.2,.2), nclass=50, col=3, main="Histogram of Randomly
Generated Data for Means")
abline(v=c(ci.u,ci.l))
#---Bootstraping the Median
var.boot.1 <- function(x,i){median(y1[i])}
boot.1<-boot(dat1,var.boot.1,1000)
boot.1
out.1 <- boot(dat1,var.boot.1,1000)
ci.var.1<-boot.ci(out.1,type="perc")
ci.var.1
hist(out.1$t)
#---Bootstraping the Mean
var.boot.2 <- function(x,i){mean(y1[i])}
boot.2<-boot(dat1,var.boot.2, 1000)
boot.2
out.2 <- boot(dat1,var.boot.2, 1000)
ci.var.2<-boot.ci(out.2,type="perc")
ci.var.2
hist(out.2$t)
#--Bootstrapping the Sample Median--1

ns<-1000
res<-numeric(ns)
for (i in 1:ns) {
res[i] <- median(sample(y1, replace=T))
se.b<-sqrt(var(res))
}
se.b
quantile(res, p = c(0.025, 0.975))
par(mfrow=c(1,1))
hist(res)
qqnorm(res)

#-Bootstrapping a Trimmed Mean

tm <- mean(y1, trim = 0.10)
nsamp <- 1000
res <- numeric(nsamp)
for (i in 1:nsamp) {
  res[i] <- mean(sample(y1, replace = TRUE), trim=0.10)
}
hist(res)
abline(v = tm, lty = 4)
sd(res)
quantile(res, p = c(0.05, 0.95))


#-------Bootstrap degree
```

```
#---Bootstraping the variance

var.boot <- function(x,i){var(d[i])}
boot<-boot(dat1,var.boot,1000)
out <- boot(dat1,var.boot,1000)
out
ci.var<-boot.ci(out,type="perc")
ci.var
hist(out$t)

#---Bootstraping the Median
var.boot.1 <- function(x,i){median(d[i])}
boot.1<-boot(dat1,var.boot.1,1000)
boot.1
out.1 <- boot(dat1,var.boot.1,1000)
ci.var.1<-boot.ci(out.1,type="perc")
ci.var.1
hist(out.1$t)
#---Bootstraping the Mean
var.boot.2 <- function(x,i){mean(d[i])}
boot.2<-boot(dat1,var.boot.2, 1000)
boot.2
out.2 <- boot(dat1,var.boot.2, 1000)
ci.var.2<-boot.ci(out.2,type="perc")
ci.var.2
hist(out.2$t)

#----End

dat2 <- dat1[,c("ln_degree", "ln_Freq", "ff")]
x1<-as.matrix(dat2$ln_degree)
y1<-as.matrix(dat2$ln_Freq)
x2<-as.matrix(dat2$ln_degree)
y2<-as.matrix(dat2$ff)
class(x1)

plot(x1,y1,xlim=range(c(x1,x2)),ylim=range(c(y1,y2)),col="red")
points(x2,y2,col="blue")

#compling vector of new variables
object<-data.frame(BC,BeC,CC, dC,evcent1, dcomp, googpr.old1)
file_cc<-file("C:\\Users\\Philipmunyua\\Google Drive\\nanoHUB research\\02 analysis\\04
develop level _test spatial\\02 input data\\centrality1219.csv","w")
write.csv(object, file_cc)
close(file_cc)

#Merging data sets
object1<-read.csv("centrality1219.csv",header=TRUE)
colnames(object1)[1] <- "Dev_Name" #Renaming a column
class(object1)
newdev<-merge(dev, object1, by="Dev_Name") #Merging developer network with the new
variables by the common variable "Dev Name"
newdev$X<-NULL
class(newdev)

file_dev<-file("C:\\Users\\Philipmunyua\\Google Drive\\nanoHUB research\\02 analysis\\04
develop level _test spatial\\02 input data\\dev_attrib_ver1219.csv","w")
write.csv(newdev, file_dev)
close(file_dev)

##Test for powerlaw for indegree (d=degree and x1=log_indegree)
#Fitting Power Law (p.239)
#fitting a power-law distribution
#d-degree
powerlawfit<-power.law.fit(d, dmin=NULL, start=2, force.continuous=FALSE,
implementation=c("plfit", "R.mle"))
powerlawx<-as.data.frame(powerlawfit)
#x1-log_indegree
```

```
powerlawfit1<-power.law.fit(x1, x1min=NULL, start=2, force.continuous=FALSE,
implementation=c("plfit", "R.mle"))
powerlawx1<-as.data.frame(powerlawfit)

###Timeline descriptives
#Assortivity
V(graph)$foo <- sample(1:3, replace=TRUE, vcount(graph))
assort<-assortativity.nominal(graph, types=V(graph)$foo)

#clusters
#calculates the maximal (weakly or strongly) connected components of a graph
isclus<-is.connected(graph, mode=c("weak", "strong")) #decided whether the graph is
weakly or strongly connected

#Diameter
#calculates the length of the longest geodesic
getdiam<-get.diameter(graph, directed=TRUE, unconnected=TRUE, weights=NULL) # retunrs a
path with actual diameter
getdiam1<-as.data.frame(getdiam)

#Dyad Census
dyads<-dyad.census(graph)
dyads1<-as.data.frame(dyads)

#Graph density
#Density is the ratio of the number of edges (links) and the number of possible edges
density<-graph.density(graph, loops=FALSE)
density1<-as.data.frame(density)

#Average nearest neighbor degree
# calculates  the average nearest neighbor degree of the given vertices and the same
quantity in the function of the vertex degree
avneigh<-graph.knn(graph, vids=V(graph),weights=NULL)
avneigh1<-as.data.frame(avneigh)

#Neighborhood of graph vertices
#finds nodes that are not farther than a given limit from another fixed node
(neighborhood of the node)
neigh.size<-neighborhood.size(graph, 1, nodes=V(graph), mode=c("all","out","in")) #
calculates the size of neighborhood
neigh.size1<-as.data.frame(neigh.size)

#Reciprocity of graphs
recipro<-reciprocity(graph, ignore.loops=TRUE, mode=c("default", "ratio"))
recipro1<-as.data.frame(recipro)

#Shortest Path
#Calculates the shortest paths between vertices
shortpath<-shortest.paths(graph, v=V(graph),mode=c("all","out","in"),
                          weights=NULL,
algorithm=c("automatic","unweighted","dijkstra","bellman-ford","johnson"))
getshortpath<-get.shortest.paths(graph, 2, to=V(graph), mode = c("out", "all",
                                                                 "in"), weights = NULL,
output=c("vpath", "epath", "both"), predecessors = FALSE, inbound.edges = FALSE)
getallshortpath<-get.all.shortest.paths(graph, 2, to = V(graph), mode = c("out",
                                                                          "all", "in"),
weights=NULL)
avshortpath<-average.path.length(graph, directed=TRUE, unconnected=TRUE)
avshortpath1<-as.data.frame(avshortpath)
pathlenngthhist<-path.length.hist (graph, directed = TRUE)

#Transivity or clustering coefficient
#A measure of the probability that the adjacency nodes of a node are connected (also
known as clustering coefficient)
clustcoeff<-transitivity(graph, type=c("undirected", "global", "globalundirected",
                                       "localundirected", "local", "average",
"localaverage",
                                       "localaverageundirected", "barrat", "weighted"),
vids=NULL,
```

```
                          weights=NULL, isolates=c("NaN", "zero"))
clustcoeff1<-as.data.frame(clustcoeff)

#Triad Census
triads<-triad.census(graph)


##----------ERGM Models

ERGM1<-gmat1~mutual
ERGM2<-gmat1~mutual+transitive
ERGM3<-gmat1~mutual+istar(3)+transitive
ERGM4<-gmat1~mutual+gwidegree(2.5, fixed=TRUE)
ERGM5<-gmat1~edges+mutual
ERGM6<-gmat1~edges+mutual+transitive
ERGM7<-gmat1~edges+mutual+istar(3)+transitive
ERGM8<-gmat1~edges+mutual+gwidegree(2.5, fixed=TRUE)

ERGM.Model.1<-ergm(ERGM1)
summary(ERGM.Model.1)
mcmc.diagnostics(ERGM.Model.1)
gof.1<-gof(ERGM.Model.1)
summary(gof.1)
plot(gof.1)
ERGM.Model.2<-ergm(ERGM2)
summary(ERGM.Model.2)
gof.2<-gof(ERGM.Model.2)
summary(gof.2)
plot(gof.2)

ERGM.Model.3<-ergm(ERGM3)
summary(ERGM.Model.3)
plot(ERGM.Model.3$sample, ask=FALSE)
gof.3<-gof(ERGM.Model.3)
plot(gof.3)
summary(gof.3)
ERGM.Model.4<-ergm(ERGM4)
summary(ERGM.Model.4)
gof.4<-gof(ERGM.Model.4)
plot(gof.4)
summary(gof.4)
mcmc.diagnostics(ERGM.Model.4)
ERGM.Model.5<-ergm(ERGM5)
summary(ERGM.Model.5)
gof.5<-gof(ERGM.Model.5)
plot(gof.5)
summary(gof.5)
mcmc.diagnostics(ERGM.Model.5)

ERGM.Model.6<-ergm(ERGM6)
summary(ERGM.Model.6)
gof.6<-gof(ERGM.Model.6)
plot(gof.6)
summary(gof.6)
mcmc.diagnostics(ERGM.Model.6)

ERGM.Model.7<-ergm(ERGM7)
summary(ERGM.Model.7)
gof.7<-gof(ERGM.Model.7)
plot(gof.7)
summary(gof.7)
mcmc.diagnostics(ERGM.Model.7)

ERGM.Model.8<-ergm(ERGM8)
summary(ERGM.Model.8)
gof.8<-gof(ERGM.Model.8)
plot(gof.8)
summary(gof.8)
mcmc.diagnostics(ERGM.Model.8)
```

```
#Anova test for models
Anova12<-anova(ERGM.Model.1,ERGM.Model.2)
Anova12
Anova13<-anova(ERGM.Model.1,ERGM.Model.3)
Anova13
Anova14<-anova(ERGM.Model.1,ERGM.Model.4)
Anova14
```

VITA

VITA

NAME OF AUTHOR: Philip Munyua

PLACE OF BIRTH: Meru, Kenya

DEGREE AWARDED:

PhD in Spatial Econometrics and Information Systems, Purdue University, West Lafayette, IN-USA, 2016

MS in Applied Economics, Purdue University, West Lafayette, IN-USA, 2012

M.A in Climate and Society, Columbia University, New York, NY-USA, 2010

M.Sc. in Applied Economics, University of Nairobi, Nairobi-Kenya 2009

BSc. in Mathematics, University of Nairobi, Nairobi-Kenya 2001

PROFESSIIONAL EXPERIENCE:

Research Assistant, Center for Open Digital Innovation, IN-USA 2014-2016.

Graduate Research Assistant, Purdue University, West Lafayette, IN-USA, 2011-2014.

Research Assistant *in the UN Millennium Village Projects (MVP),* Earth Institute, Columbia University, New York, NY,2009-2010

Consultant, EATTA/UNEP, Nairobi, Kenya, 2008-2009.

Research Assistant *in Social-Economics Department,* Kenya Agricultural Research Institute-KARI, Embu, Kenya. 2003–2007