IN

Université du Québec en Outaouais

Doctorat en sciences et technologies de l'information (STI)

Département d'informatique et d'ingénierie

Impact de la disparition d'un nœud sur la structure d'un réseau social

Par

Dyah Anggraini

Thèse présentée au Département d'informatique et d'ingénierie

Pour l'obtention du grade de

PHILOSOPHIAE DOCTOR (PhD)

En sciences et technologies de l'information

Jury d'évaluation

Président du jury	Mme Nadia Baaziz, Université du Québec en Outaouais (UQO)
Directeur de recherche	Mme Rokia Missaoui, UQO
Examinateur interne	Ana-Maria Cretu, UQO
Examinateur externe	Morad Benyoucef, University of Ottawa
Examinateur externe	Idrissa Sarr, Université Cheikh Anta Diop

Gatineau, Québec, Canada, 2015

© Droits réservés de Dyah Anggraini, 2015

Université du Québec en Outaouais

31 AOUT 2016

Bibliothèque

Résumé

Impact de la disparition d'un nœud sur la structure d'un réseau social

Par

Dyah Anggraini

De nos jours, l'analyse de réseaux sociaux intéresse plusieurs chercheurs de divers domaines dans un grand spectre de problématiques comme la centralité (importance) d'un noeud, l'évolution de réseaux, l'identification de communautés, etc. Les noeuds centraux sont ceux qui sont reliés à d'autres noeuds du réseau d'une manière intensive ou critique soit parce qu'ils ont plusieurs voisins, jouent le rôle d'intermédiaire ou sont les plus proches du reste des nœuds. Ainsi, l'élimination de ces noeuds centraux peut se propager et aboutir à une désintégration du réseau ou un ralentissement dans la diffusion de l'information.

Les deux principales contributions de cette thèse sont la proposition de solutions aux problèmes suivants : (i) réorganisation d'un réseau suite à la disparition d'un noeud, et (ii) la prédiction de l'effet cascade après l'élimination d'un ou plusieurs nœuds du réseau.

Pour le premier problème, nous proposons une solution qui identifie un substitut au noeud éliminé chaque fois que ce dernier joue un rôle important dans le réseau comme leader, médiateur ou témoin avec un score élevé de la centralité de degré, d'intermédiarité ou de proximité respectivement. Des liens entre le substitut et d'autres nœuds du réseau sont ensuite ajoutés selon les besoins de l'application en vue de maintenir la stabilité et la connectivité du réseau.

Pour le second problème, nous proposons deux variantes d'un modèle de cascades qui identifient les noeuds candidats à une élimination lorsqu'un ou plusieurs nœuds disparaissent d'un réseau avec liens orientés et valués. Notre solution propose une formule de poids non triviale à appliquer aux nœuds voisins de ceux éliminés pour prédire la disparition possible de ces voisins. Une analyse empirique des approches proposées sur des réseaux aussi bien réels que synthétiques a été effectuée en comparaison avec quelques méthodes existantes et a fait ressortir les forces et faiblesses de nos solutions.

Université du Québec en Outaouais

Doctorat en sciences et technologies de l'information (STI)

Département d'informatique et d'ingénierie

Impact of Node Disappearance on a Social Network Structure

By

Dyah Anggraini

A thesis presented to the University of Québec in Outaouais as a fulfillment of the thesis requirement for the degree of Doctor of Philosophy in Science and Information Technologies

Committee Members

Chairman :	Nadia Baaziz, Université du Québec en Outaouais (UQO)
Research Advisor :	Rokia Missaoui, UQO
Internal Examiner :	Ana-Maria Cretu, UQO
External Examiner :	Morad Benyoucef, University of Ottawa
External Examiner :	Idrissa Sarr, Université Cheikh Anta Diop

Gatineau, Québec, Canada, 2015

©Droits réservés à Dyah Anggraini, 2015

Abstract

Impact of Node Disappearance on a Social Network Structure

By

Dyah Anggraini

Social network analysis is nowadays attracting many researchers from different domains to study a large spectrum of issues such as node centrality (importance), network evolution, community identification and so on. Central nodes are the ones that are linked to other nodes in the network in an extensive or critical manner because either they have many neighbors, or play the role of mediator or are very close to the other nodes. Therefore, the removal of these central nodes may propagate and lead to a disintegration of the network or a slowness in the information flow.

In this thesis, we will cover two main topics: (i) network reorganization following a node disappearance, and (ii) cascading removal following the deletion of one or many nodes from the network. For the first issue, we propose an approach that seeks for a substitute node whenever the disappearing node plays an important role in the network either as a leader, mediator or witness with a high score of degree, betweenness, and closeness centrality, respectively. Links between the substitute and other nodes in the network are then added to maintain stability of the network and avoid disconnectivity.

For the second issue, we propose a cascading effect method to identify the nodes that could disappear from a weighted and oriented network once one or many nodes are deleted. Our solution integrates a non-trivial formula for computing the weight of the neighbors of deleted nodes in order to predict the possible disappearance of those neighbors.

An empirical study of the proposed approaches on real and synthetic networks has been conducted. Their comparison with a few existing methods has highlighted their strengths and weaknesses.

Acknowledgments

This thesis would not have been possible without the support and the help of many people during my PhD studies.

I owe my deepest gratitude to Prof. Rokia Missaoui for her patience, great guidance and advices. She keeps me motivated and inspired to complete my thesis. Without her support, help and suggestions, this thesis would not have been possible. It is an honor working with Madame Missaoui who first gave me an opportunity to stay at LARIM (Laboratoire de Recherche en Information Multimédia) as an intern from September to December 2008.

I would like to thank the former postdoctoral fellows at LARIM, namely Amir Razavi, Idrissa Sarr and Elsa Negre for their help and numerous insightful comments and discussions about a few topics of my thesis. I am also grateful to many professors of the Department of CS and Engineering for their encouragement and confidence.

I would like to acknowledge the financial support of my supervisor through her grant from the Natural Sciences and Engineering Research Council of Canada (NSERC).
My gratitude goes to the jury members: Professors Nadia Baaziz, Ana-Maria Cretu, Morad Benyoucef and Idrissa Sarr who accepted to evaluate my thesis.

I would like to thank Directorate General of Higher Education, Ministry of Education and Culture of the Republic of Indonesia together with Yayasan Pendidikan Gunadarma that granted the scholarship for the Doctorate Program at the Université du Québec en Outaouais, Québec, Canada.

I am deeply indebted to the Rector of Gunadarma University, Prof. Dr. E.S. Margianti, SE., MM and Vice Rector II of Gunadarma University, Prof. Suryadi H.S., SSI., MMSI who gave me the opportunity and trust to continue my study at Université du Québec en Outaouais. Without their trust and their financial support, this thesis would not have been possible.

I would like to thank Prof. Dr. Sarifuddin Madenda and Dr. Peni Sawitri from Gunadarma University who believed in me from the very beginning of my enrollment in the PhD program. They gave me the chance to pursue my studies at Université du Québec en Outaouais.

My thanks go also to my dearest friends, Mbak Rita Gupta and her family as well as Yunita Dwiyanasari and her family for their support during the final year of my study.

Lastly but not the least, I would like to show my deep gratitude to my family for their endless love, incredible support, help and understanding.

List of Publications

Missaoui, R., Negre, E., Anggraini, D., & Vaillancourt, J. (2013). Network restructuring after a node removal. *International Journal Conference of Web Engineering and Technology (IJWET), 8*(1), 4-26.

Razavi, A. H., Anggraini, D., Missaoui, R., Vaillancourt, J., & Talbi, M. (2014). Modeling and Predicting Cascading Phenomenon over Social Network. *Social Network Analysis and Mining (SNAM), 4(1), Springer, DOI 10.1007/s13278-014-0233-1.*

Dedication

I would like to dedicate this thesis to my supervisor "Rokia Missaoui" and to my dearest family (Papa, Mama, Mas dan Adek Adek yang terkasih dan tercinta).

Table of Contents

Résumé ii

Abstract iv

Acknowledgments v

List of Publications vii

Dedication viii

Table of Contents ix

List of Figures xii

List of Tables xv

List of Algorithms xvii

1.Introduction 1

2. Background 4

2.1 Social networks 4

2.1.1 Directed and undirected network 6

2.1.2 Subgraph, Simple graph and Multigraph 6

2.1.3 Random, Small-world and Scale-free network 7

2.1.4 Walk, path, shortest path and geodesic distance 8

2.1.5 One-mode and Two-mode Data 9

2.1.6 Bridge 9

2.1.7 Weighted Graph 10

2.1.8 Degree 10

2.2 Centrality measures 11

2.3 Network measures 20

2.4 Conclusion 22

3. Related work 23

3.1 Link Prediction 23

3.2 Cascading Removal 27

3.3 Community Detection 30

4. Network Restructuring After Node Disappearance 33

4.1 Introduction 33

4.2 Node Centrality and Link Options 34

4.3 Our Proposed Approach 39

4.3.1 Role 39

4.3.2 Algorithms 41

4.3.3 Additional Functions 44

4.3.4 Community Detection 45

4.4 Empirical Study 45

4.4.1 First Series 46

4.4.2 Second Series 50

4.5 Conclusion 69

5. Cascading Removal 70

5.1 Introduction 70

5.2 Proposed Approach 70

5.2.1 Network features 71

5.2.2 Cascading strategies 72

5.2.3 An illustrative example 73

5.3 Procedure Validation 76

5.3.1 Cascading triggers 77

5.3.2 Removal prediction accuracy 78

5.4 Empirical Study 79

5.4.1 Data set 79

5.4.2 Environment 81

5.4.3 Experimental results and discussion 82

5.5 Conclusion 86

6. Conclusion and Future Work 88

Appendix 90

References 94

List of Figures

Figure 1. A Modified version of KITE network as a matrix 5

Figure 2. A modified version of KITE network as a graph 5

Figure 3. (a). Undirected network, (b). Directed network, (c). Loop and directed multigraph 6

Figure 4. Walks in a network (Hanneman & Riddle, 2005) (a) Undirected and (b) Directed network 8

Figure 5. A bridge (Easley & Kleinberg, 2010) 10

Figure 6. Eigenvector centrality values 14

Figure 7. Authority and hub 18

Figure 8. The core 19

Figure 9. Left: KITE network. Right: the centrality scores 35

Figure 10. (a) The initial network. (b) After deleting node *B* and related ties. (c) After forming a clique with nodes *A, D, E*, and *G* that were neighbors of *B* 36

Figure 11. (a) The new network AFTER deleting node D and related ties. (b) AFTER using the link option *IMP* with ε = 25%. (c) AFTER using the link option *OLD* 37

Figure 12. Execution time (in milliseconds) 47

Figure 13. F-measure of the substitutes for the two data sets according to ε 48

Figure 14. Network perturbation: COAUTHOR, ε = 20%, and with or without substitutes 49

Figure 15. Network perturbation: COAUTHOR, ε = 20%, and link options 49

Figure 16. Variation of the number of edges for the *karate club network* with a non-critical node removal 54

Figure 17. Variation of link number for the *karate club network* with a critical node removal 54

Figure 18. Processing time for the *karate club network* with a non-critical node removal 55

Figure 19. Processing time for the *karate club network* with a critical node removal 55

Figure 20. Variation of centrality measures for the *karate club network* with a non-critical node removal 56

Figure 21. Variation of centrality measures for the *karate club network* with a critical node removal 56

Figure 22. Variation of topological measures for the *karate club network* with a non-critical node removal 57

Figure 23. Variation of topological measures for the *karate club network* with a critical node removal 57

Figure 24. Variation of link number for *collaboration network* with a non-critical node removal59

Figure 25. Variation of link number for *collaboration network* with a critical node removal 59

Figure 26. Processing time for *collaboration network* with a non-critical node removal 60

Figure 27. Processing time for *collaboration network* with a critical node removal..... 60

Figure 28. Variation of centrality measures for *collaboration network* with a non-critical node removal 61

Figure 29. Variation of centrality measures in case of A/A or IF for *collaboration network* following a critical node removal 62

Figure 30. Variation of centrality measures using OLD or IMP for *collaboration network* after a critical node removal 62

Figure 31. Variation of topological measures for *collaboration network* with a non-critical node removal 63

Figure 32. Variation of topological measures for *collaboration network* with a critical node removal 63

Figure 33. Variation of link number for the *Barabási network* with a non-critical node removal 64

Figure 34. Variation of the link number for the *Barabási network* with a critical node removal.. 65

Figure 35. Processing time for the *Barabási network* with a non-critical node removal..... 65

Figure 36. Processing time for *Barabási network* with a critical node removal 66

Figure 37. Variation of centrality measures for Barabási network with a non-critical node removal 66

Figure 38. Variation of centrality measures for the *Barabási network* with a critical node removal 67

Figure 39. Variation of topological measures for *Barabási network* with a non-critical node removal 67

Figure 40. Variation of topological measures for the *Barabási network* with a critical node removal 68

Figure 41. Network example 74

Figure 42. The accuracy of cascading removal prediction of synthetically generated networks via SVM classifier 84

Figure 43. The accuracy of cascading removal prediction of synthetically generated networks via Bayesian Logistic Regression classifier 84

Figure 44. The accuracy of cascading removal prediction of real networks via SVM Classifier.. 85

Figure 45. The accuracy of cascading removal prediction of real networks via Bayesian Logistic Regression Classifier 85

List of Tables

Table 1. Network types 7

Table 2. Scale-free network examples (Barabási & Bonabeau, 2003) 8

Table 3. The degree and neighbors of the modified KITE Network 10

Table 4. Geodesic distance values for the modified version of KITE network 15

Table 5. Centrality scores of the modified KITE network 16

Table 6. The hub and authority values 19

Table 7. The similarity of nodes from modified KITE network 20

Table 8. Proximity measures (Liben-Nowell & Kleinberg, 2003) 25

Table 9. Centrality measures for the new network SN' after deleting D and using *IMP* to create new links 38

Table 10. Comparison between non critical versus critical node removal for the *karate club network* 58

Table 11. Non critical versus critical node removal for *collaboration network* 64

Table 12. Non-critical versus critical node removal for the *Barabási network* 68

Table 13. The weights of network example 75

Table 14.The computed weight of the candidate nodes for removal 76

Table 15. Number of trigger nodes and their neighbors for the three networks 80

Table 16. Execution time for real-life social networks (in milliseconds) 80

Table 17. The accuracy of cascading removal prediction for synthetically generated networks 82

Table 18. The accuracy of cascading removal prediction for synthetically generated networks using Bayesian Logistic Regression classifier 82

Table 19. The accuracy of cascading removal prediction for real networks via SVM classifier... 83

Table 20. The accuracy for real networks via Bayesian Logistic Regression classifier 83

List of Algorithms

Algorithm 1. Predicting the structure of a network 42

Algorithm 2. Finding the substitute of a deleted node 43

1.Introduction

Social network analysis has received much attention during the last decades and has been studied according to different approaches and techniques (Knoke & Yang, 2008; Wasserman & Faust, 1994) by researchers from different disciplines (*e.g.,* Physics, Social Science, Computer Science, Mathematics). A social network is a set of nodes and links where nodes can be individuals, organizations, objects or events while links represent interactions or relationships. Based on its number of interactions with the rest of the network, a node can have a more or less central role. The network is represented as a graph and several measures can be used to evaluate the centrality of a node within this structure. For instance, the degree, betweenness, and closeness centrality measures (Wasserman & Faust, 1994) are among the measures frequently used to assess the role of nodes within a network. The features of a network as a whole are also important. The density and the diameter of a network are such features.

Our thesis covers two linked topics, namely network restructuring after a node disappearance, and cascading effect of node removal. We rely on node importance and network topology to study these two issues.

A social network is a dynamic structure where nodes and links can appear and disappear for many reasons. Therefore, it is trivial to understand that the disappearance of a central node does not have the same effects as the disappearance of a marginal node. For example, the disappearance of a central node can create a cascading effect and lead to a disintegration of the network or slowness in information dissemination.

The first topic exploits the role played by the nodes in the network to predict the future structure of the related graph after the disappearance of one or even many nodes. It uses the network topology to possibly find a substitute to the deleted node and add new links between identified nodes in order to maintain the stability of the network. To reach our goal, we improved the solution initially described in Negre *et al.* (Negre, Missaoui, & Vaillancourt, 2011) and we analyze the impact of network restructuring by capturing the overall picture of the network before and after the update using known network measures such as clustering coefficient, network diameter efficiency and so on. An empirical comparison of our approach against two related ones is also provided.

Another important issue that we handle is the prediction of a deletion in cascade (snowball/cascade effect) when a set of nodes disappear from a directed and weighted network. From real-life situations, the disappearance of nodes or links can propagate and create a considerable impact on the network. Their disappearance may also spread from one

node to another across the network creating a cascading behavior. The cascading phenomenon is mainly analyzed either by identifying the most influential nodes according to their features in the network, by quickly detecting the phenomenon or by targeting a minimum set of nodes that could maximize the spread of influence or minimize the propagation of a rumor or an outbreak. For example, an organization may predict the attrition of a group of its employees when a leader leaves the organization or when a tie between a boss and one of his employees vanishes.

To summarize, the goal of this research work is twofold: (i) to propose a solution (with variants) towards updating the network structure once one node is deleted so that the new structure fulfills some conditions (*e.g.,* slight changes in the overall network structure) and the deleted node is replaced with an existing one whenever it is central, and (ii) to predict the nodes that could disappear following a node deletion. In general, given a network (of computers or individuals or objects), what happens when an entity is removed from the network (*e.g.,* employee's retirement or resignation, computer breakdown)? What will the new structure of the network be? Is there any potential substitute that can play the role of the removed node? What are the appropriate links that more likely need to be created to avoid the fragmentation of the network into disconnected sub-networks? Which node(s) could be affected by the removal of a node or a set of nodes? To that end, we will be concerned with the following issues:

1. How to maintain the network connected without drastically changing the overall structure of the network?
2. How to predict a propagation of a deletion to many nodes once a set of nodes disappear?

We assume in this research work that (i) the topology a network contains or reveals important information that could be exploited for data analysis and network structure changes, and (ii) node centrality is an important measure in social network analysis (SNA) as stated by many leading researchers.

Our work is new *w.r.t.* the literature because the first contribution on node removal and substitution deals with both the identification of a substitute to a deleted node (if it is central) and network reorganization to maintain the structure connected and stable. The previous studies were dedicated either to updating (mainly adding) links or finding a substitute. The second contribution about cascading removal is original since we exploit and improve well-known propagation models that have been initially proposed for influence spread to our own needs, namely the identification of nodes that could disappear once a set of nodes are deleted. While the existing propagation models rely on randomness and lead to non

deterministic solutions, our approach is based on a formula that computes the importance/weight of a node in a weighted and directed graph.

The whole thesis work is important because it models real-life situations (e.g., such as customer/employee attrition and manager retirement) and proposes realistic solutions.

The rest of the document is organized as follows. The basic concepts and definitions are introduced in Chapter 2. Chapter 3 presents related work. In Chapter 4, we present our approach towards finding a substitute to a deleted node and adding new links. Our solution towards predicting cascading effects of node removal is discussed in Chapter 5. Finally, Chapter 6 concludes the thesis and summarizes our future work.

2. Background

A social network is defined as a set of nodes where some pairs of the nodes are connected by one link (tie, arc, edge) or more. Nodes represent objects, actors, people, or organizations while links express collaborations, communications or interactions. A link might be directed or undirected and possibly weighted.

In this chapter we will introduce some basic notions that will be used throughout the thesis. The definitions and representations of social networks are discussed in Subsection 2.1 while a few node centrality scores are considered and explained in Subsection 2.2. The discussions of network measures are found in Subsection 2.3.

2.1 Social networks

A network is generally a large, complex and dynamic structure. The network may contain millions of nodes and links, and have a complex structure with various connectivity forms (*e.g.* isolated components, intermediate structure and tightly knit structure (Knoke & Yang, 2008)). A network with isolated components contains nodes that are not linked directly or indirectly to other nodes. In an intermediate network structure, some of the nodes are intensively connected while other ones are sparse, and lastly in a tightly knit network, the network is very dense. Nodes and links may have attributes. For example, a person has attributes such as his status, religion, and occupation. A network is very dynamic since nodes and links may be added or removed from the network over time due to many reasons, such as a new employee joins or leaves a company.

There are many categories of networks: social, communication/transportation, biological, information networks and so on. In social networks, there are friendship networks like Facebook and Myspace, professional networks like LinkedIn, and media sharing networks like Flickr or YouTube. Communication/transportation networks cover for example airline or road networks while information networks make reference to web graphs or co-authorship networks.

There are at least two ways to represent a network, either by graphs or matrices (Hanneman & Riddle, 2005). In graph theory, we represent a social network SN as a couple $\langle V, E \rangle$ where V is a set of N nodes (vertices) and E is a set of M interactions (edges) between vertices. In a matrix representation, the network is expressed in a tabular display as an adjacency matrix. An adjacency matrix $M(N,N)$ represents matrix M with dimensions of N rows and N columns and N^2 cells displaying information about the relations between the N

nodes. M_{ij} is 1 if there is a link between nodes i and j, and 0 otherwise. Two nodes are adjacent if there is a link between them.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
2	1	0	1	1	0	0	0	1	0	0	1	0	0	0	0
3	1	1	0	1	1	0	0	0	1	0	0	0	0	0	0
4	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0
5	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0
6	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0
7	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
8	0	1	0	1	0	0	0	0	1	1	0	0	0	0	0
9	0	0	1	1	1	0	0	1	0	1	0	0	0	0	0
10	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0
11	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0
12	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
13	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
14	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1
15	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Figure 1. A Modified version of KITE network as a matrix

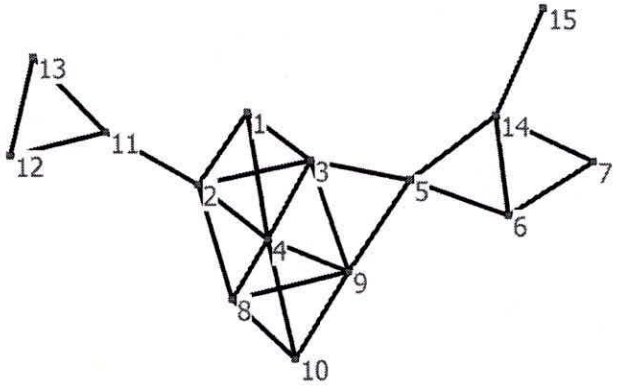
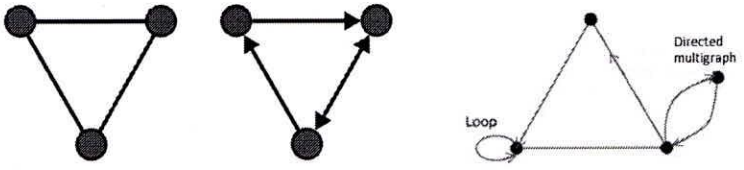


Figure 2. A modified version of KITE network as a graph

2.1.1 Directed and undirected network

A network can either be directed or undirected. In an undirected network $SN = \langle V, E \rangle$ where $E \subseteq V \times V$, the line (or edge/link) has no arrowhead (no orientation). The relation is mutual (*e.g.,* coworkers). The edge connects a pair of nodes without considering the *origin* or *destination* node. Every $e_{ij} \in E$ has distinct nodes $i \neq j$ as endpoints such that nodes i and j are said to be adjacent in the network. Given two nodes i and j, in an undirected network, the edge that connects these nodes is denoted by $\{i, j\}$ or ij or e_{ij} as illustrated in Figure 3a. In an undirected graph there is no loop (edge between a node and itself).

In a directed network (digraph), $SN = \langle V, E \rangle$ with $E \subseteq V \times V$, the edge (or arc) connects the nodes in a directed way. A single-headed arrow $e_{ij} \in E$ indicates a direct edge from the node at the tail (*origin*) node i to the other node at the head (*destination*) node j. The edge is represented by the notation e_{ij} where $e_{ij} \neq e_{ji}$. As shown in Figure 3b, an edge with arrowheads on both ends indicates two directed edges which suggest mutuality or reciprocity.



(a). Undirected (b). Directed (c). Loop and Multigraph

Figure 3. (a). Undirected network, (b). Directed network, (c). Loop and directed multigraph

2.1.2 Subgraph, Simple graph and Multigraph

A subgraph $SN' = \langle V', E' \rangle$ is a subset or a part of nodes and edges from a larger graph $SN = \langle V, E \rangle$, such that $V' \subseteq V$ and $E' \subseteq E$. An example of a subgraph is a community. A community in a graph is a subgraph that has higher connection of edges between its members and lower connection of edges from its members outside the subgraph.

A simple graph $SN = \langle V, E \rangle$, is an undirected graph without multiple edges and loops. A graph which contains several or parallel edges between nodes is called a multigraph. Each edge between these nodes can hold optional data or attributes. Table 1 below shows different types of networks.

Table 1. Network types

Type of Graph	Edges	Multi Edges allowed?	Loops allowed?
Simple graph	Undirected	No	No
Multigraph	Undirected	Yes	No
Directed graph	Directed	No	Yes
Directed multigraph	Directed	Yes	Yes

2.1.3 Random, Small-world and Scale-free network

A random network of Erdös and Rényi is a network created by a single isolated node and the edges (links) are added between pairs of nodes selected uniformly at random. Erdös and Rényi give variations of their network model. The most common random network is $SN_{N,p}$ in which every possible link between nodes i and j is present with an independent probability p. In a such model, all graphs with N vertices and M edges have the probability of $p^M(1-p)^{R-M}$ where $R = \frac{1}{N} N(N-1)$ (M. E. Newman, 2003). A random network exhibits a small average of path length along with a small clustering coefficient (Hopkins, 2010).

In social networks, nodes are generally connected to other nodes through very short paths. This phenomenon is known as small-world phenomenon or the six degrees of separation (Easley & Kleinberg, 2010). The first significant experimental research on the small-world phenomenon was carried out by the social psychologist Stanley Milgram. Small-world networks are characterized by a small average of shortest path length and a large clustering coefficient. In Orman and Labatut (Orman & Labatut, 2009), the authors mentioned that a network has the small-world property, if for a fixed average degree, its average distance (*e.g.,* the length of the shortest paths) between pairs of nodes increases logarithmically with the number N of nodes. The authors think that this property can be inferred as the efficiency of propagation, which means that the spread of information in the network remains quite fast even if the size of the network becomes higher

A scale-free network is a network that follows a power law degree distribution, at least asymptotically. Such a structure is characterized by two main features, growth and preferential attachment. Growth indicates that the number of nodes in the network increases in time while preferential attachment means that new nodes tend to connect to vertices with large degree. The percentage $P(k)$ of nodes in the network that have k connections to other nodes is proportional to $k^{-\gamma}$ for a high value of k:

$P(k) \sim k^{-\gamma}$ where γ is a parameter with a value generally in the range $2 < \gamma < 3$ (Choromanski, Matuszak, & Miekisz, 2013; R. Cohen, Havlin, & Avraham, 2006). Table 2 below gives a few examples of scale-free networks.

Table 2. Scale-free network examples (Barabási & Bonabeau, 2003)

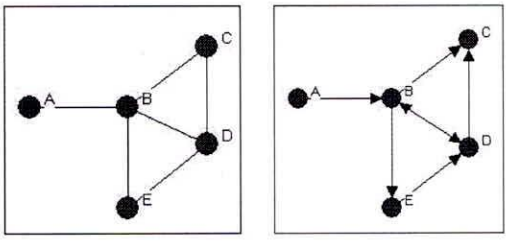
Network	Nodes	Links
Cellular metabolism	Molecules involved in burning food for energy	Participation in the same biochemical reaction
Internet	Routers	Optical and other physical connections
Protein regulatory	Proteins that help to regulate a cell's activities	Interaction among proteins
Research collaboration	Scientists	Co-authorship of papers
World Wide Web	Webpages	URLs

We used *Barabási network* as a scale-free network in our empirical study.

2.1.4 Walk, path, shortest path and geodesic distance

The most general form of connections between two nodes in a network is called a walk. A walk in a graph *SN* of length *k* from node *i* to *j* is a sequence of nodes and edges.

A trail is a walk with no repeated edge, *i.e.,* it does not pass over the same edge more than once. A trail might visit the same node twice, but only if it comes and goes from a different edge each time. A path is a walk with no repeated nodes, except the first node might be the same as its last. A cycle is a closed path since it starts and ends with the same node.



(a) (b)

Figure 4. Walks in a network (Hanneman & Riddle, 2005) (a) Undirected and (b) Directed network

In Figure 4(a), there are many walks of different lengths. One can see for example one walk of length two (nodes A, B, and C), and one walk with length three (A, B, D and C). All trails are walks but not all walks are trails. In our example, connections from Node A, B, D, B and C are a walk but not a trail since the edge between nodes B and D is used twice. In Figure 4(b), there are three paths with different lengths connecting nodes A and C. The paths are: A, B and C, A, B, D and C, and lastly nodes A, B, E, D and C. One possible trail from A to C is A, B, E, D, B and C.

Given graph $SN = \langle V, E \rangle$, a shortest path between two nodes i and j is the path of minimum length between these two nodes. The length of a path in a weighted graph is the sum of the weights of its edges. The geodesic distance between two nodes is the number of edges in a shortest path connecting these nodes.

There are different kinds of algorithms for computing the shortest paths: (i) automatic, (ii) unweighted, (ii) Dijkstra, (iv) Bellman-Ford, and (v) Johnson algorithm. Dijkstra algorithm chooses the node with the shortest distance that has not been visited yet, and then computes the distances through this node to each one of its unvisited neighbors. For a network of $|V|$ nodes and $|E|$ edges, the complexity of Dijkstra's algorithm is $O(|V| \, log \, |V| + |E| \, log \, |V|)$, which is $O(|E| \, log \, |V|)$ when the network is connected because $|V|$ is $O(|E|)$. The complexity of Bellmon-Ford Algorithm is $O(|V||E|)$ while Johnson's algorithm complexity is $O(|V|2 \, log \, |V| + |V| \, |E|)$. We chose Dijkstra's algorithm for its time efficiency and memory space requirement to compute the shortest path in an undirected network.

2.1.5 One-mode and Two-mode Data

In a network of one mode data there is only one type of node and one type of link. For example in an airline network, the nodes represent the airports and the links represent the flight routes. In two mode data there are two types of nodes. An example of two mode data can be a set of researchers that attend scientific events.

2.1.6 Bridge

A bridge is a direct link between nodes that would otherwise disconnect the graph into clusters or communities. In Figure 5 below, removing the edge between node A and B will disconnect the network into two parts.

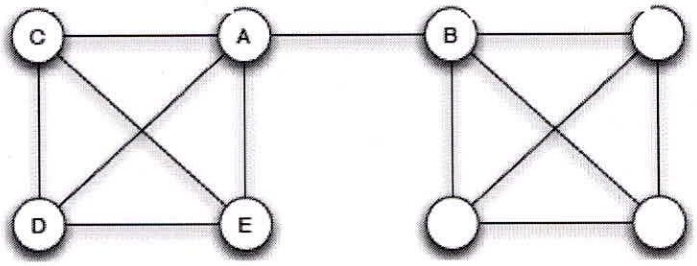


Figure 5. A bridge (Easley & Kleinberg, 2010)

2.1.7 Weighted Graph

A weighted graph is a graph for which each edge $e \in E$ has an associated weight w. The edge weight may represent the strength of the link, the distance between two vertices or the cost. For example in a flight route network, the weight of an edge could represent the distance in miles between endpoints of the airports or the frequency of the route during a given time period.

2.1.8 Degree

The degree of a node i, denoted by $d(i)$, indicates its importance in terms of the number of its neighbors or equivalently the number of edges $e_{ij} \in E$ incident to it. The neighborhood of a node i in an undirected network, represented by $N(i)$ is the set of nodes adjacent to node i.

Table 3. The degree and neighbors of the modified KITE Network

Node i	$d(i)$	N(i)
1	3	2, 3, 4
2	5	1, 3, 4, 8, 11
3	5	1, 2, 4, 5, 9
4	6	1, 2, 3, 8, 9, 10
5	4	3, 6, 9, 14
6	3	5, 7, 14
7	2	6, 14
8	4	2, 4, 9, 10
9	5	3, 4, 5, 8, 10
10	3	4, 8, 9
11	3	2, 12, 13
12	2	11, 13
13	2	11, 12
14	4	5, 6, 7
15	1	14

In a directed network, a node i has incoming and outgoing edges connected to it. *Indegree* of node i is the number of directed edges coming to i. It is denoted by $deg^-(i)$. *Outdegree* of i, denoted by $deg^+(i)$ is the number of edges directed out from node i. Therefore, the total number of directed edges in the network is given by the following formula:

$$|E| = \frac{\sum_{i \in V} deg^-(i) + \sum_{i \in V} deg^+(i)}{2}$$

2.2 Centrality measures

Centrality measures are the most widely used measures in graph theory (Borgatti, 2005; Freeman, 1977). These measures are among the main and commonly used conceptual tools for exploring and investigating the role of nodes in social networks. For example, the degree, betweenness and closeness centrality measures are frequently applied to find the importance of a node (Wasserman & Faust, 1994) or a group of nodes (Everett & Borgatti, 2005) in the network. There are also other measures to capture the role of nodes such as eccentricity, PageRank, hub and authority value, transitivity and eigenvector (Knoke & Yang, 2008; Wasserman & Faust, 1994).

Centrality measures are not the same when applied to directed or undirected networks. The degree centrality of an undirected and unweighted network is the extent to which a node has the highest number of interactions. Given a social network $SN = \langle V, E \rangle$, where V is a set of N nodes (vertices) and E is a set of M interactions (edges) between vertices, the degree of node i is the number of all the existing edges e_{ij} such that $i \neq j$. $e_{ij} = 1$ if there is an edge between nodes i and j and 0 otherwise. In undirected networks, the degree centrality is:

$$d(i) = \sum_{i \neq j}^{N} e_{ij}$$

For directed networks, there is a distinction between indegree and outdegree measures as seen before.

A standardized or normalized measure for all the centrality measures was suggested in Wasserman and Faust (Wasserman & Faust, 1994) to allow the comparison of networks with different sizes. The normalized degree centrality for node i in the network SN is: $C_D^{SN}(i) = \frac{d(i)}{N-1}$, where $d(i)$ is the degree of node i and N is the total number of nodes in the network.

The node centrality measure was extended by Everett and Borgatti (Everett & Borgatti, 2005) to a set of nodes. The degree centrality of a group G of nodes represents the number of nodes outside G which are linked to elements in G. The normalized version of the group degree centrality is given below:

$$C_D^{SN}(G) = \frac{|N(G)|}{N-|G|},$$

where $N(G)$ is the set of nodes which do not belong to group G but are adjacent to an element of the group and |G| stands for the cardinality of the set G.

The betweenness centrality expresses the amount of the flow control that a node (or a group of nodes) has over the interaction of other nodes within the network. According to Krebs (Krebs, 2011), a node with a high betweenness centrality has a huge effect on all the information exchange that runs in the network and can act as a cut point. Therefore, removing this node may disconnect the network. Betweeneess centrality is computed as $C_B^{SN}(i) = \sum_{i \neq j \neq k} \frac{p_{jk}(i)}{p_{jk}}$ where p_{jk} indicates the shortest path between nodes j and k and $p_{jk}(i)$ is the number of shortest paths between nodes j and k that cross node i. The normalized betweenness centrality for node i in directed networks is computed as:

$$C_B^{SN}(i) = \frac{\sum_{i \neq j \neq k} \frac{p_{jk}(i)}{p_{jk}}}{(N-1)(N-2)}$$

In undirected networks, this centrality measure is $C_B^{SN}(i) = \frac{2 \times \sum_{i \neq j \neq k} \frac{p_{jk}(i)}{p_{jk}}}{(N-1)(N-2)}$.

For a group of nodes, the betweenness centrality is computed as follows:

$$C_B^{SN}(G) = \frac{\sum_{j<k} \frac{p_{jk}(G)}{p_{jk}}}{(N-|G|)(N-|G|-1)}.$$

In undirected networks, the betweenness centrality of a group is calculated by

$$C_B^{SN}(G) = \frac{2 \times \sum_{j<k} \frac{p_{jk}(G)}{p_{jk}}}{(N-|G|)(N-|G|-1)},$$

where p_{jk} indicates the shortest path between nodes j and k and $p_{jk}(i)$ (respectively $p_{jk}(G)$) is the number of shortest paths between nodes j and k that cross node i (respectively G). N is the total number of the nodes within the network.

The closeness centrality of a node indicates how this node is close to the other nodes in the network and hence determines how fast the information circulates from that node to other reachable nodes. Closeness centrality of a network is computed as

$$C_C^{SN}(i) = \frac{1}{\sum_{j=1, i \neq j}^{N} d_{ij}}.$$

A normalized closeness centrality of node i is calculated as follows:

$$C_C^{SN}(i) = \frac{N-1}{\sum_{j=1, i \neq j}^{N} d_{ij}},$$

where d_{ij} is the shortest distance between nodes i and j.

The study in Rochat (Rochat, 2009) extends the closeness centrality measure from connected to unconnected networks. A main drawback of closeness centrality measure is its inadequacy in unconnected networks. The distance between two nodes belonging to different components is infinite, which makes the sum of the distances in their network infinite too and therefore its inverse is equal to zero. The authors propose an index of harmonic centrality, which is briefly described in Newman (M. E. J. Newman, 2003) and instead of the inverted sum of the distances, it is defined as the sum of inverted distances: $\sum_{i \neq j} \frac{1}{d_{ij}}$. Harmonic centrality computation avoids situations where an infinite distance outweighs the others. The index is normalized with N - 1, and the index of harmonic centrality is defined by:

$$C_H^{SN}(i) = \frac{1}{N-1} \sum_{j \neq} \frac{1}{d_{ij}} .$$

In Csardi and Nepusz (Csardi & Nepusz, 2006), the authors propose to replace the infinite distance between two nodes that belong to two different components by the number of nodes in the network. It is defined as the largest geodesic distance in a network of N nodes with a length equal to $N-1$. The closeness centrality then can be generalized as:

$$C_\alpha^{SN}(i) = \frac{N-1}{\sum_{i \neq j} d_{ij} + L\alpha}$$

with node j selected in the same connected component as node i, $N = |V|$, L is the number of nodes unconnected to node i and α is a constant greater than or equal to the diameter of the network.

Everett and Borgatti (Everett & Borgatti, 2005) define group closeness centrality as "the normalized inverse sum of distance from the group to all nodes outside the group". This measure is computed as follows:

$$C_C^{SN}(G) = \frac{\sum_{i=1}^{N} [C_C^{SN}(N^*) - C_C^{SN}(i)]}{[(N-2)(N-1)]/(2N-3)} ,$$

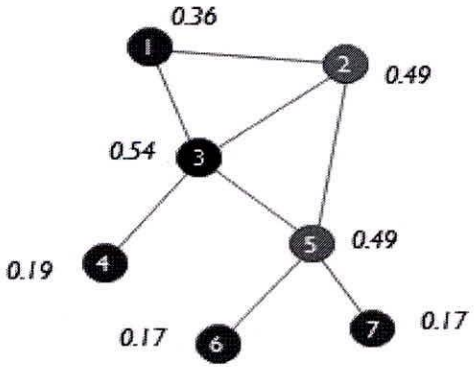
where $C_C^{SN}(N^*)$ is the highest closeness centrality value in the network.

Calculating the betweenness and closeness centralities of N nodes in a network with M edges assumes the computation of the shortest paths between all pairs of nodes. Many algorithms compute the betweenness and closeness centrality in $O(N^3)$. However, since the density of the network has an impact on computing the time complexity values, it has been shown that the calculation of betweenness centrality requires $O(N + M)$ space and runs in $O(N*M)$ time for unweighted networks using the algorithm proposed by Brandes (Brandes, 2001), while the time complexity for closeness centrality computation is $O(N*M + N^2 \log N)$ as shown in Okamoto *et al.* (Okamoto, Chen, & Li, 2008). The study in Zhuge and Zhang

(Zhuge & Zhang., 2010) shows that the complexity of degree centrality computation can reach $O(N^2)$ in dense networks but becomes $O(M)$ for sparse ones.

To summarize the centrality measures explained above, these measures highlight the importance of nodes in the network according to different perspectives. For instance, degree centrality identifies an individual that has a great influence on its adjacent nodes (*i.e.,* its neighbors), betweenness centrality emphasizes the highest value to nodes through which node information is more likely to get through. Closeness centrality shows how quickly a node is able to reach (and spread information through) other nodes in the network.

Another centrality measure used to express the importance of nodes is eigenvector centrality. This measure is used to determine which node is connected to the most connected or linked nodes. Eigenvector centrality of node i is proportional to the sum of all the eigenvector centralities of all nodes that are adjacent to node i. In our example (see Figure 6), node 3 has the highest eigenvector centrality (1.53) followed by node 2 (1.39) and node 5 (1.37).



Network A (Cheliotis, 2010)

From network A:

Node 1 : Connected to node 2 and 3 (1.03)

Node 2 : Connected to node 1, 3 and 5 (1.39)

Node 3 : Connected to node 1, 2, 4 and 5 (1.53)

Node 4 : Connected to node 3 (0.54)

Node 5 : Connected to node 2, 3, 6 and 7 (1.37)

Node 6 : Connected to node 5 (0.49)

Node 7 : Connected to node 5 (0.49)

Figure 6. Eigenvector centrality values

Beside eigenvector centrality, the eccentricity measure of a node in the network is related to closeness centrality and hence determines also the importance of nodes. The eccentricity $\varepsilon(i)$ of a node i in a connected network is the maximum geodesic distance between node i to any other node j in the network and hence is expressed as $\varepsilon(i) = max\{d_{ij},\ j \in V\}$. Node i with the smallest $\varepsilon(i)$ is the closest one from the rest of nodes. In a disconnected network, all nodes are identified to have infinite maximum distance (B.West,

2001). The minimum eccentricity is called radius *rad*(*SN*) while the maximum eccentricity is the diameter of the network.

Table 4 represents the geodesic distance value of each node for the modified version of KITE network described by David Krackhardt (Krebs, 2011) as shown in Figure 2 where 15 actors are linked through 26 connections.

Table 4. Geodesic distance values for the modified version of KITE network

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	1	1	1	2	3	4	2	2	2	2	3	3	3	4
2	1	0	1	1	2	3	4	1	2	2	1	2	2	3	4
3	1	1	0	1	1	2	3	2	1	2	2	3	3	2	3
4	1	1	1	0	2	3	4	1	1	1	2	3	3	3	4
5	2	2	1	2	0	1	2	2	1	2	3	4	4	1	2
6	3	3	2	3	1	0	1	3	2	3	4	5	5	1	2
7	4	4	3	4	2	1	0	4	3	4	5	6	6	1	2
8	2	1	2	1	2	3	4	0	1	1	2	3	3	3	4
9	2	2	1	1	1	2	3	1	0	1	3	4	4	2	3
10	2	2	2	1	2	3	4	1	1	0	3	4	4	3	4
11	2	1	2	2	3	4	5	2	3	3	0	1	1	4	5
12	3	2	3	3	4	5	6	3	4	4	1	0	1	5	6
13	3	2	3	3	4	5	6	3	4	4	1	1	0	5	6
14	3	3	2	3	1	1	1	3	3	3	4	5	5	0	1
15	4	4	3	4	2	2	2	4	4	4	5	6	6	1	0

To illustrate the usage of centrality measures on our example, we provide the following table computed by the UCINET software (see www.analytictech.com/ucinet/) where actors *3, 4,* and *5* are the central actors due to their highest centrality scores.

Table 5. Centrality scores of the modified KITE network

Node	Degree Centrality	Betweenness Centrality	Closeness Centrality	Eigenvector Centrality	Eccentricity
1	0.214	0	0.424	0.397	4
2	0.357	0.372	0.482	0.513	4
3	0.357	0.322	**0.518**	0.557	**3**
4	**0.429**	0.068	0.467	**0.663**	4
5	0.286	**0.440**	0.483	0.304	4
6	0.214	0.060	0.368	0.107	5
7	0.143	0	0.285	0.050	6
8	0.286	0.037	0.437	0.485	4
9	0.357	0.147	0.467	0.550	4
10	0.214	0	0.389	0.389	4
11	0.214	0.264	0.368	0.136	5
12	0.143	0	0.280	0.041	6
13	0.143	0	0.280	0.041	6
14	0.286	0.203	0.378	0.111	5
15	0.071	0	0.280	0.026	6

There are different types of sub-structures that can be found in a network: clique, chain, N-clique, N-clan, k-core etc. A chain is a sequence of edges while a k-core is a connected maximal subgraph whose minimum degree is greater than or equal to k. A clique is a maximal complete sub-network SN' of at least three nodes (Wasserman & Faust, 1994) in which all nodes are connected to one another. From Figure 2, we obtain eight cliques.

1. Nodes 1, 2, 3 and 4
2. Nodes 2, 4 and 8
3. Nodes 4, 8, 9 and 10
4. Nodes 3, 4 and 9
5. Nodes 3, 5 and 9
6. Nodes 5, 6 and 14
7. Nodes 6, 7 and 14
8. Nodes 11, 12 and 13

Another important measure that computes the centrality in the network is information centrality C^I (Fortunato, Latora, & Marchiori, 2004; Sun, 2014). It is based on the idea of information propagation through the network (Latora & Marchiori, 2001, 2003). The information or communication in a network flows along the shortest paths (geodesic

distances) between nodes while the efficiency $\varepsilon_{i,j}$ in the communication between nodes i and j is equal to the inverse of the shortest path length between nodes i and j (d_{ij}).

The efficiency of network SN is used to compute the information centrality and is defined as:

$$E(SN) = \frac{\sum_{i \neq j \in SN} \epsilon ij}{N(N-1)} = \frac{1}{N(N-1)} \sum_{i \neq j \in SN} \frac{1}{d_{ij}}$$ (Latora & Marchiori, 2001, 2003)

The value of $E(SN)$ varies in the range of $[0, 1]$. For unconnected networks, when there is no path between nodes i and j, the distance $d_{ij} = +\infty$ and therefore $e_{ij} = 0$. Fortunato *et al.* (Fortunato et al., 2004) use this measure to quantify the importance of an edge in the network.

The information centrality (C_k^I) of an edge k is computed as:

$$C_k^I = \frac{\Delta E}{E} = \frac{E(SN) - E(SN_k')}{E(SN)}$$ (Fortunato et al., 2004)

where $k = 1, 2, \ldots M$, and SN_k' is a new network that is obtained by removing edge k from network SN. According to Fortunato *et al* (Fortunato et al., 2004), the formula works also when SN_k' is unconnected.

Information centrality is used by Fortunato *et al* to find communities in a network. The general algorithm works as follows:

1. Compute information centrality for each edge in the given network.
2. Remove the edge(s) with the highest centrality value from the network.
3. Perform an analysis and examination of the network components before and after edge removal.
4. Return to Step 1 until all the edges with the highest centrality are removed and the network is disconnected.

The authors expect that the edges that link communities are those with the highest information centrality value, whereas the edges inside the communities have low information centrality. As indicated before, the computation of shortest paths can be performed with a breadth-first search algorithm in $O(N*M)$ (Brandes, 2001; M. E. J. Newman, 2001) where N and M are the number of nodes and the number of edges respectively. The calculation of the information centrality for all edges in a network takes $O(N*M^2)$ time (Fortunato et al., 2004).

PageRank is another important measure used in social network analysis. It has emerged as the dominant link analysis model that was originally developed for indexing web pages since 1998 (Liu, 2014). PageRank is determined through incoming links and is applied in directed networks like a retweet network on Twitter (Tsvetovat & Kouznetsov, 2011) and Google (Liu, 2014) . It is a method for ranking the importance of Web pages and relies on the independent nature of the web by using its huge link structure as indicator of an individual

page value or quality. A page is important if it is cited by other important pages. The higher incoming links a page *i* receives, the more prestige that page *i* has. According to rank prestige, the importance of page *i*, *i.e.*, its PageRank score, is the sum of the PageRank scores of all pages that point to *i*. Since a page may link too many other pages, its prestige score should be shared. For example, if page *i* has multiple outgoing links (*outdegree* > 1), then its PageRank contribution is equally divided by all the links of the target. $PR_C(i) = \sum_{(j,i)\in E} \frac{PR(j)}{Outdegree(j)}$ (Liu, 2014). *PR(i)* is the PageRank of page *i*, and *PR(j)* is PageRank of the page that points to page *i* (*e.g.*, citations), and *Outdegree(j)* is the number of outgoing links for page *j*. In Chapter 5 we will exploit PageRank to compute the weight of a directed edge. The other measures given below will also be used.

PageRank gives scores to every page in order to help users (including Web communities) find relevant and useful information in key pages. It is related to pages with many outgoing links (hub) and pages with many incoming links (authorities). In other words, hubs are the web pages that point to a large number of authorities, *i.e.*, they have a large number of outgoing links, and authorities are the web pages pointed by a large number of hubs, *i.e.*, they have a large number of incoming links (Caldarelli, 2007).

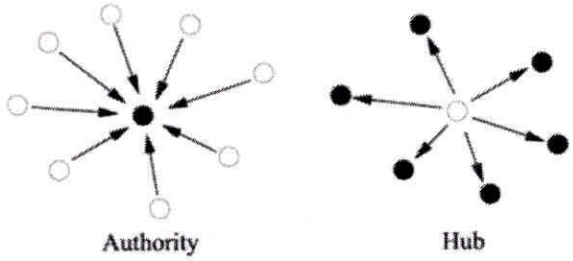


Figure 7. Authority and hub

Based on the modified KITE network, the hub and authority of each node can be seen in Table 6. Since our modified KITE network is an undirected network, the values of hub and authority are the same.

Table 6. The hub and authority values

Node	Hub	Authority
1	0.281	0.281
2	0.363	0.363
3	0.393	0.393
4	**0.469**	**0.469**
5	0.215	0.215
6	0.075	0.075
7	0.035	0.035
8	0.343	0.343
9	0.389	0.389
10	0.275	0.275
11	0.096	0.096
12	0.029	0.029
13	0.029	0.029
14	0.079	0.079
15	0.018	0.018

The centralization measure is used to explore whether a network is centralized or dispersed. It is based on the calculation of centrality values between nodes. A network that depends on few highly connected nodes (*e.g.,* the result of preferential attachment) will have few very high connected nodes and many nodes with low degree (Cheliotis, 2010). In addition to centralization, many networks have core tightly-knit connected users that are critical for connecting a much larger set of peripheral nodes (Cheliotis, 2010).

Cores in a network can be identified as nodes with high degree values and high joint degree distribution since high degree nodes have a tendency to connect to other high degree nodes.

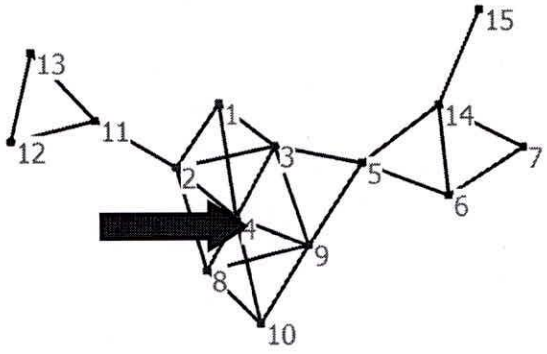


Figure 8. The core

From Figure 8 and Table 6, we see that the core of the modified KITE network is node 4.

Beside all the centrality measures discussed above, there is another measure called similarity which has been viewed as a function of structural and role equivalence (Friedkin & Johnsen, 1997). The purpose is to compare two nodes and compute a single number which evaluates their similarity using commonly known measures such as the Pearson correlation measure.

Table 7. The similarity of nodes from modified KITE network

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1.000	0.640	0.640	0.539	0.030	-0.300	-0.234	0.426	0.318	0.133	0.133	-0.234	0.234	-0365	-0.158
2	0.640	1.000	0.278	0.501	-0.184	-0.433	-0.337	0.030	0.350	0.318	-0.284	0.101	0.101	-0.527	-0.228
3	0.640	0.278	1.000	0.501	0.030	-0.058	-0.337	0.501	0.278	0.318	-0.058	-0.337	-0.337	-0.184	-0.228
4	0.539	0.501	0.501	1.000	0.051	-0.507	-0.395	0.693	0.501	0.539	-0.141	-0.395	-0.395	-0.617	-0.267
5	0.030	-0.184	0.030	0.051	1.000	0.272	0.640	-0.083	0.030	0.030	-0.365	-0.284	-0.284	0.133	0.433
6	-0.300	-0.433	-0.058	-0.507	0.272	1.000	0.677	-0.365	-0.058	-0.300	-0.300	-0.234	-0.234	0.133	0.433
7	-0.234	-0.337	-0.337	-0.395	0.640	0.677	1.000	-0.284	-0.337	-0.234	-0.234	-0.182	-0.182	0.527	0.677
8	0.426	0.030	0.501	0.693	-0.083	-0.365	-0.284	1.000	0.426	0.778	-0.030	-0.284	-0.284	-0.444	-0.192
9	0.318	0.350	0.278	0.501	0.030	-0.058	-0.337	0.426	1.000	0.640	-0.433	-0.337	-0.337	-0.184	-0.228
10	0.133	0.318	0.318	0.539	0.030	-0.300	-0.234	0.778	0.640	1.000	-0.300	-0.234	-0.234	-0.365	-0.158
11	0.133	-0.284	-0.058	-0.141	-0.365	-0.300	-0.234	-0.030	-0.433	-0.300	1.000	0.677	0.677	-0.365	-0.158
12	-0.234	0.101	-0.337	-0.395	-0.284	-0.234	-0.182	-0.284	-0.337	-0.234	0.677	1.000	1.000	-0.284	-0.123
13	-0.234	0.101	-0.337	-0.395	-0.284	-0.234	-0.182	-0.284	-0.337	-0.234	0.667	1.000	1.000	-0.284	-0.123
14	-0365	-0.527	-0.184	-0.617	0.133	0.133	0.527	-0.444	-0.184	-0.365	-0.365	-0.284	-0.284	1.000	0.000
15	-0.158	-0.228	-0.228	-0.267	0.433	0.433	0.677	-0.192	-0.228	-0.158	-0.158	-0.123	-0.123	0.000	1.000

A similarity value of 1 indicates perfect connection (*i.e.,* perfect homophily) and a value of -1 indicates a perfect negative connection, *i.e.,* perfect heterophily (Chase & Borgatti). Homophily is the tendency to connect nodes with similar characteristics (*e.g.,* status, beliefs, age, hobby, etc.). Homophily leads to the construction of clusters.

In Chapter 5, we also exploit the transitivity of nodes, which is the indication of strong connection between nodes in a network (Cheliotis, 2010). If there is a link between nodes i and j, and one between nodes j and k, then in a transitive way nodes i and k will be connected. The transitivity property is computed by a transitivity coefficient also known as clustering coefficient (Strogatz & Watts, 1998). The higher the clustering coefficient, the more likely we detect an edge between two nodes which are connected to a third one.

2.3 Network measures

Network measures (Kaiser, 2008; Strogatz & Watts, 1998) are metrics that allow the quantification of a network as a whole and are helpful for comparing and classifying a set of graphs (*e.g.,* samples of an initial graph) or when one network has undergone some

topological changes (*e.g.*, node/edge insertion or removal). There are many measures used to capture the overall picture of a network, including the following ones: diameter of the network, its density, the degree distribution and the clustering coefficient.

The diameter of the network gives information about the size of the network by computing the maximum length of the geodesic distance. Such a distance between two nodes is defined as the number of edges in a shortest path connecting them. In the case of unconnected entities, the geodesic distance is either set to ∞ or computed as the greatest distance in the network augmented by 1 (Hanneman & Riddle, 2005).

Network density (Δ) helps describe the shape of the network as a whole (Hanneman & Riddle, 2005) and represents the extent to which the network is complete. A complete network is one where all the nodes are adjacent to each other (*i.e.*, they form a clique). The density of a network is the ratio of the total number of links over the number of all possible links between nodes. It is a common measure to verify how well connected a network is. Let N be the number of nodes and M the number of edges, the density of an undirected network is computed as follows:

$$\Delta = \frac{2M}{N(N-1)}$$

For a directed network, the formula is:

$$\Delta = \frac{M}{N(N-1)}$$

The (node) degree distribution provides the number of nodes with degree k for $k = 0, 1, \ldots, p$. The degree of each node might not be the same. This variability is described by the degree distribution $P(k)$ within the network as defined below, where N_k stands for the number of nodes having degree k in a network of N nodes.

$$P(k) = \frac{N_k}{N}$$

The clustering coefficient of a network (Strogatz & Watts, 1998) is the average value of the local clustering coefficients of its N nodes. $C = \frac{1}{N}\sum_{i=1}^{N} C_i = \frac{1}{N}\sum_{i=1}^{N} \frac{2\Gamma_i}{d(i)(d(i)-1)}$, where $d(i)$ is the degree of node i while Γ_i represents the number of edges between the neighbors of node i. An adjusted (and more reliable) clustering coefficient is proposed in Kaiser (Kaiser, 2008) where N in C is replaced with the number of nodes that have at least two neighbors.

For example, the diameter, density and clustering coefficient of the network shown in Figure 2 are 6, 0.2476 and 0.662 respectively.

2.4 Conclusion

In this chapter we have introduced the basic notions and the fundamental concepts related to graph theory and social network analysis that will be used through the rest of the thesis. In the first part we discussed the notions of graphs, their structure and properties while in the second part we introduced centrality measures and lastly, we presented network measures. These concepts will be extensively used in the thesis.

3. Related work

A social network is a dynamic structure (Ben-Naim & Krapivsky, 2007; Easley & Kleinberg, 2010; Toivonen et al., 2009) where nodes and links can be added and/or deleted due to different reasons. For example in a friendship network, a person (node) can be removed from the network because she/he is not loyal to the other members. In an airline network, a flight that uses a given route between two airports may drop that route to use another one for security reasons. Many social network evolution models have been proposed. However, they are mainly based on adding nodes or links.

Since our thesis deals with three related topics, we will dedicate three subsections to the related literature about (i) link prediction, (ii) cascading removal, and (iii) community detection. The last topic is important because it can be used as a preprocessing step in network reorganization or even disintegration.

3.1 Link Prediction

Link prediction is an important issue in social network analysis and aims at forecasting the new links to be added in an existing network. Links are often created using the triadic closure and the focal closure (see (Kumpula, Onnela, Samaraki, Kaski, & Kertész, 2007; Toivonen et al., 2009)) through non-deterministic approaches which are generally limited to the geodesic neighborhood of the node. We recall that the triadic closure is the probability that node B is linked to C knowing that A is already linked to B and C. The focal closure is the probability that A and B that share an interest could interact while the geodesic neighborhood of A is the set of nodes located at a geodesic distance (*i.e.,* the shortest path) from A.

There are numerous applications of link prediction in many research areas such as information retrieval, e-commerce (*e.g.,* recommendation systems), bioinformatics (*e.g.,* protein interaction), and criminal networks. Al Hasan and Zaki (AlHasan & Zaki, 2011) conducted a survey on link prediction techniques with respect to their prediction performance, complexity, scalability, and generalization ability. They consider three types of link prediction models: (i) traditional approaches, (ii) probabilistic approaches, and (iii) linear algebraic approaches. In the first group, supervised learning techniques are mainly used. The probabilistic approaches model the joint-probabilities among the entities in a network mainly through Bayesian probabilistic models while in the last group the similarity of nodes in a network is computed using rank-reduced similarity matrices.

The studies in Liben-Nowell and Kleinberg (Liben-Nowell & Kleinberg, 2003) and Tylenda *et al.* (Tylenda, Angelova, & Bedathur, 2009) focus on predicting new links but exclusively in static social networks. In Liben-Nowell and Kleinberg (Liben-Nowell & Kleinberg, 2003), the authors developed an approach for link prediction based on measuring the "proximity" of a node in the network. The authors formulated the link prediction task as: Given a social network $SN = \langle V, E \rangle$ where $e_{ij} \in E$ represents a connection between nodes i and j at time t. For time $t \leq t'$, let us assume that $SN'[t, t']$ is a subnetwork of SN that is restricted to the links with time stamps between t and t'. In link prediction with a supervised training system, let $[t_0, t_0']$ be a training interval and $[t, t']$ a test interval, where $t_0' \leq t_1'$. The link prediction task consists of producing a set of links that are not present in $SN'[t_0, t_0']$ but are predicted to appear in $SN'[t_1, t_1']$. The information for predicting future interaction between nodes is hidden inside the network structure, thus the idea is to identify the appropriate proximity measure that effectively predicts the new links that will likely occur in the future by assuming that two close nodes have greater probability to be linked. Let $\Gamma(x)$ be a set of neighbors of node x in the network. Nodes x and y are more likely to build a tie in the future if the sets of neighbors $\Gamma(x)$ and $\Gamma(y)$ largely overlap. Three categories of link prediction methods are defined and experimented in their studies:

1. Those based on a node neighborhood such as graph distance, common neighbors measure, Jaccard's coefficient and *Adamic/Adar* distance.
2. Those based on the computation of the ensemble of all paths like Katz measure, Hitting time and SimRank
3. Higher-order methods like low-rank approximation, unseen bigrams and clustering techniques.

The authors conducted intensive empirical tests to estimate the accuracy of predicting the new structure based on a set of proximity measures. They found that the Katz measure performs well while some very simple measures like common neighbors and *Adamic/Adar* distance provide a good accuracy. We recall below some of these proximity measures.

Table 8. Proximity measures (Liben-Nowell & Kleinberg, 2003)

Jaccard's coefficient	$\frac{\lvert \Gamma(x) \bigcap \Gamma(y) \rvert}{\lvert \Gamma(x) \bigcup \Gamma(y) \rvert}$
Adamic/Adar	$\sum_{z \in \lvert \Gamma(x) \cap \Gamma(y) \rvert} \frac{1}{\log \lvert \Gamma(z) \rvert}$
$Katz_{\beta}$	$\sum_{\ell=1}^{\infty} \beta^{\ell} \cdot \lvert path_{x,y}^{\langle \ell \rangle} \rvert$ Where : • $\lvert path_{x,y}^{\langle \ell \rangle} \rvert$:= {path of length exactly ℓ from x to y} • Weighted : $\lvert path_{x,y}^{\langle 1 \rangle} \rvert$:= number of collaborations between x and y • Unweighted : $\lvert path_{x,y}^{\langle 1 \rangle} \rvert$:= 1 if and only if x and y collaborate.

Another study shows that supervised learning can be usefully exploited for link prediction (see AlHasan *et al.*(AlHasan, Chaoji, Salem, & Zaki, 2006)). Among six classification algorithms (with variants) tested on two large bibliographic co-authorship graphs (BIOBASE and DBLP) using both topological features (*e.g.,* shortest path) and domain-specific features (*e.g.,* keyword match count), the empirical experiments suggest that support vector machine (SVM) procedure has the best predictive power while naive Bayes algorithm has a bad performance, and the shortest path is the best topological feature. In a more recent work, (Romero & Kleinberg, 2010) analyze and formalize the process of directed closure in the follower network of Twitter. The study shows that there are many directed triangles (a "feed-forward" structure on three nodes), in which users initially follow users closer to micro-celebrities than they are, before following the micro-celebrities directly. This might indicate that both central nodes used in our current work on node removal and substitution (see the next chapter) are important in creating new links: leaders usually have many neighbors while witnesses offer better closeness to the rest of nodes in a network. If either type of node disappears, the triadic closure (triangle) process studied in (Romero & Kleinberg, 2010) could collapse.

Tylenda *et al.* (Tylenda et al., 2009) take into account the evolution of the network over time by integrating the weights of edges into the existing link prediction methods. The authors also examine a new method that assesses the presentation of different prediction algorithms in their capacity to rank nodes from a neighborhood of a selected node.

The issue of node addition and deletion was studied in Ben-Naim and Krapivsky (Ben-Naim & Krapivsky, 2007). The authors studied networks with a biased growth so that a new node is inserted in the network at a given rate while a node is deleted with a smaller frequency in such a way that its parent node is newly linked to its immediate descendants. The proposed growth network model seems relevant to a few specific networks such as biological ones.

The deletion of a node leads to the deletion of the adjacent links and the substitution of the removed node is needed as suggested in Toivonen *et al.* (Toivonen et al., 2009). The authors propose two models for adding new links in the network: link additions that are based on the network structure and those in which links are created according to node attributes.

Hussain and Ahmed (Hussain & Ahmed, 2010) handle the problem of identifying the node in the network which will replace the deleted one. To that end, they use a Bayesian approach to predict which node can replace the deleted node in the network, by first computing the posterior probability of each node in the network. The best substitute of the removed node is the one that has the closest posterior probability to that of the deleted node. They focus on a network where nodes are organized in a hierarchical way so that the substitute needs to be at the same level as the deleted node (*e.g.,* when a manager leaves the company for some reasons, she or he should be replaced by another manager in the company). Unfortunately, link management is not discussed in this study.

The objective of the work in Missaoui *et al.,* Negre *et al.*, and Sarr & Missaoui (Missaoui, Negre, Anggraini, & Vaillancourt, 2013; Negre et al., 2011; Sarr & Missaoui, 2013) is to exploit the topology of the network to develop an approach towards (i) finding (a) substitute(s) of the deleted node, and (ii) adding the links to avoid the disintegration of the network. These three studies use a role-based approach that investigates the role played by the nodes in the network to predict the new structure of the social network once one of its nodes disappears. The algorithms work in two steps: (i) predict which node in the network will replace the deleted node X if the latter has a specific role, and (ii) predict the new links that will appear in the network. In Negre *et al.* (Negre et al., 2011), the authors consider two roles of the nodes: the leader and the mediator with their corresponding measure: the degree centrality and the betweenness centrality. Once a node and its related links are deleted from the network, there is a search for one or possibly many substitutes for the deleted node if the latter happens to be a leader or a mediator. Then, links are added based on the user's choice. When the deleted node has no role, no substitute is sought because this node is not critical enough to be replaced by another one, but links between the neighbors of the deleted node are added. The work in Negre *et al.* (Negre et al., 2011) has been improved in Missaoui *et al.* (Missaoui et al., 2013) and will be discussed in detail in Chapter 4 of this thesis.

The study of Sarr and Missaoui (Sarr & Missaoui, 2013) describes a new approach which exploits the role played by a given node to both estimate the impact of its disappearance on the *information flow* and conduct minimal network changes to maintain and even improve the quality of information flow despite the disappearance of the node. The authors divide the nodes into two categories: (i) critical nodes, *i.e.,* the ones having the highest (degree, betweenness and closeness) centrality measures and representing the leader,

mediator and witness respectively, and (ii) non-critical nodes, *i.e.,* the ones that might have high centrality measures but not the highest value. The authors divide the non-critical nodes into two distinct groups: fingers (nodes that are slightly less important than critical ones) and followers (nodes with lower centrality measures). Based on these categories, the authors reconstruct the network and manage node removal by selecting a substitute in case of critical node removal and adding new links in a parsimonious way. When the deleted node is non critical, links are added mainly to maintain the network connected and the quality of information flow. The authors propose two new algorithms called JOAN (Join-Only-Alike-Node) and JOAN-C. The former is called when new links are added in the network when a non-critical node is removed while the latter is used to find a substitute to a deleted critical node and add new links.

Sarr *et al.* (Sarr, Missaoui, & Lalande, 2013) expand their research to the removal of a set of nodes rather than only one. Deleted nodes are either critical or non-critical ones, and three kinds of node sets are considered: scattered, contiguous and hybrid groups. Moreover, two distinct cases are studied: the network as a whole and the case of identified communities. Finally, algorithms are defined to handle group disappearance according to different variants. New links are added in a parsimonious way and a possible group substitute for a leaving group is found based on the citation ''birds of a feather flock together'' and the homophily principle. As a consequence, new links and group substitute are found only between individuals that share common characteristics, and more likely neighbors of the leaving node set.

3.2 Cascading Removal

Network reliability and stability are bases of any social network. Therefore, the vulnerability of a network to various types of failures is an important topic in many subdomains of graph theory as well as social and computer networks (Blume, Easley, Kleinberg, Kleinberg, & Tardos, 2011; Huang, Vodenska, Havlin, & Stanley, 2013; Zio & Sansavini, 2011). The failures of nodes (*e.g.,* the deletion of a few nodes) may spread from one node to another one across the network and cause a disintegration of the network into disconnected parts. A node may be removed from a network due to many reasons; for instance an employee may leave one company to join another company to get a better position and trigger the departure of other employees. The above issue motivated us to focus on the vulnerability versus stability of networks following a cascading effect of node removal.

Cascading (snowball) failures normally propagate in a variety of social networks. The most tangible example of such issues can be cascade failures among financial systems during recessions (Allen & Gale, 2000) or epidemic disease in a population (Anderson & May, 1992). In social networks research area, there are studies around different aspects of cascading effects (Cui et al., 2013; Leskovec, Krause, et al., 2007; Leskovec, McGlohon, Faloutsos, Glance, & Hurst, 2007). In (Leskovec, Krause, et al., 2007), a multiple criteria approach is proposed for selecting influential nodes. Those influential nodes identify outbreaks (propagation of a virus/information) efficiently by defining optimization objectives such as detection time or contaminated population minimization. Note that cascading removal is a phenomenon that normally appears in a dynamic, complex and large social network. It starts when the disappearance of a single or a group of nodes is followed by the departure of other nodes in the network. It even tends to proceed with an increasing strength so that the further the removal propagates, the more difficult is the maintenance of network connectivity and stability. Analyzing the cascading removal effect and proper approaches to minimize its influence in a social network is an interesting and challenging topic in social networks. When an innovation or idea appears, it can either disappear quickly or create a major impact into the network. Therefore, it would be interesting to see to what extent such cascading effects are predictable by modeling their behavior within social networks.

In order to model the propagation of influence cascade including removal cascade effects, various cascade models have been developed (Chen, Wang, & Yang, 2009; Kempe, Kleinberg, & Tardos, 2003; Kleinberg, 2007). Leskovec, McGlohon *et al.* (Leskovec, McGlohon, et al., 2007) define a model that mimics real cascades and helps analyze cascading behavior and information propagation. Such a model can also be useful to predict new links, detect outliers and deal with "what-if" scenarios in networks.

Fowler and Christakis (Fowler & Christakis, 2009) have conducted an empirical study to show that cooperative and uncooperative behaviors lead to a cascading effect in human social networks. Since people tend to imitate and hence propagate the behavior of other persons, it has been empirically shown that the spread of behavior or influence over time goes up to three degrees of separation. This observation may suggest that the proposed approach described in this thesis can be executed either in three iterations or a repetitive way until the distance between one initial deleted node (trigger) and an affected node reaches the value 3.

Cui *et al.* (Cui et al., 2013) propose a new method that both optimizes outbreak prediction (rather than detection) and node selection. More precisely, the proposed solution consists to first select important nodes and further predict the outbreaks based on the cascading behavior of the preselected nodes. The authors exploit Orthogonal Sparse Logistic

Regression (OSLOR) to select a set of influential nodes that act as sensors and use these nodes to predict the nodes affected by the cascading effect. Placing the sensor as a monitor to detect the cascade behaviors has been widely applied in (Christakis & Fowler, 2010; R. Cohen, Havlin & Ben-Avraham, 2003; Krause & Guestrin, 2009). In our approach, the identified nodes are user-driven rather than pre-computed, and information spread is materialized by the deletion of the triggered identified nodes.

Authors in (Domingos & Richardson, 2002) were the first that studied the influence maximization as an algorithmic problem. Kempe *et al.* (Kempe et al., 2003) showed that the influence maximization problem can be approximated through "Linear Threshold" and "Independent" cascade models. The concept of the "Linear Threshold" model was introduced by a sociologist named Mark Granovetter (Granovetter, 1978) who was among the first researchers who proposed a model that potentially captures the influence propagation (Kempe et al., 2003) based on the use of node-specific threshold. He assumed that each node v has a random positive threshold $\theta v \leq 1$ that corresponds to some different hidden tendencies that lead it to adopt the innovation or idea from the set A_v of active neighbor nodes of vertex v. A node is defined as an active one when the node either acts as an innovator or adopter of an innovation/idea. Otherwise, the node is inactive. For every node $w \in A_v$, node v is influenced by w according to a weight $0 \leq b_{w,v}$ such that $\sum_{w \in A_v} b_{w,v} \leq 1$.

Given a random value of θ_v, an initial set A_i of active nodes at time t, then node v will also become active at time $t+1$ when $\sum_{w \in A_v} b_{w,v} \geq \theta_v$. This means that node v becomes active when at least (weighted) θ_v fraction of its neighbors are active.

According to Kempe *et al.* (Kempe et al., 2003), theoretically the simplest model for the propagation is the Independent Cascade Model. This model starts with an initial set of active nodes A_i. When a node v becomes active, it has a single chance of activating an inactive neighbor w with a probability $p_{v,w}$ at time $t+1$. The random event can be defined by tossing coin of bias $p_{v,w}$. If node v has multiple activated neighbors at time $t+1$, their attempts will run in a random manner. The process runs until no more activation is possible. In both Linear Threshold and Independent Cascade Models, the active nodes will never deactivate.

In the linear cascade model, Kempe *et al.* (Kempe et al., 2003) consider the multiplicity of edges as weights. If nodes u and v are linked through $c_{u,v}$ parallel edges, the edge e_{uv} has weight $\frac{c_{u,v}}{d(v)}$ and edge e_{vu} has weight $\frac{c_{u,v}}{d(u)}$where u and v have the degrees $d(u)$ and $d(v)$, respectively. The authors study a special case of Independent Cascade Model since the Independent Cascade Model with uniform probabilities p on edges has the property that nodes with high degree not only have the probability to influence many other nodes in a

network but also to be influenced by them. Motivated by the above argument, the authors introduce another interpretation where the edges with high degree values are assigned small probabilities p. A specific case of Independent Cascade Model, called "Weighted Cascade", is also studied by the authors. In such case, each edge from node u to node v has a probability $p = 1/d_v$ of activating node v. The authors state that for the Weighted Cascade Model and the Linear Threshold model, the expected number of neighbors that would succeed to activate node v is 1. They confirm that the optimization issue is NP-hard, then present a greedy approximation algorithm that is adaptable to the three different cascade models (Independent Cascade, Linear Threshold and Weighted Cascade model) and guarantee that the influence propagation is within (1-1/e-ε) of the optimal influence propagation, where e is the base of the natural logarithm and ε is any positive real number.

3.3 Community Detection

Identifying communities in a network is an important issue in social network analysis (see (Fortunato, 2010) for a survey) and has attracted many researchers in sociology, biology, computer science, and so on. A community is a kind of cluster where many edges link nodes of the same cluster and few edges link nodes of different clusters. A commonly used approach to find communities is based on edge betweenness centrality (Girvan & Newman, 2002)which avoids having isolated nodes but has high computational requirements. Another commonly used method is based on the modularity maximization (M. E. J. Newman, 2004) which calculates the quality of a particular clustering of a network into communities. The intuition behind modularity is that nodes inside the same community are highly connected but loosely linked to nodes outside their community. Some further optimizations have been proposed. One of them is a parameter-free and easy-to-use approach described in Chen *et al* (Chen et al., 2009).

The optimization of modularity is done through a greedy algorithm called Fast Greedy proposed by Newman (M. E. J. Newman, 2004). As a bottom-up procedure, it puts each node i in a separate community and iteratively merges together node pairs. At each step, the merge of nodes that maximizes modularity is chosen. The algorithm stops when the modularity value does not increase.

A hierarchical divisive algorithm is introduced in Newman (M. E. Newman & Girvan, 2004) which consists of removing the edges in the network in a decreasing order of their edge betwenneess scores. We recall that edge betweenness centrality is computed based on the number of the shortest paths between two given nodes. The idea behind their approach is that the links with the highest edge betweenness score are bridges between communities

and hence their removal leads to the identification of communities. The approach gives a good precision in terms of community identification, but induces a high computational time due to the need to recompute the edge betweenness score every time an edge is removed.

Another method for community detection is a spectral approach by Newman (M. E. Newman, 2006), which is a top-down hierarchical one that depends on eigenvectors of the modularity matrix. This approach works by iteratively separating the network into two components so that the modularity is maximized.

A hierarchical agglomerative algorithm called *Walktrap* was proposed by Pons and Latapy (Pons & Latapy, 2006). The algorithm employs the similarity between nodes in the network based on random walks for the merge of communities in a bottom-up way. Once the resulting dendrogram is produced, the partition that leads to the best modularity is retained. According to Fortunato (Fortunato, 2010), the computation of node similarity is based on the random walk probabilities p from one node to another node in a fixed number of steps. The number of steps is a user-defined threshold which has an impact on the produced communities.

In Raghavan *et al.* (Raghavan, Albert, & Kumara, 2007), the authors proposed a community detection method based on *label propagation*. At the beginning, each node has a unique label. Afterwards, labels are assigned iteratively to nodes in a random sequential order in such a way that nodes take the most frequent label of its neighborhood. The label reassignment converges when there is no more change in the node label. Groups of nodes that have identical nodes at convergence form communities. Although this approach is efficient and does not require user-defined parameters, it is not deterministic due to the random choice of nodes to be labeled and to the possibly large number of edges explored during the iterations.

Another community detection method has recently been proposed in Bouguessa *et al.* (Bouguessa, Missaoui, & Talbi, 2014). It is a two-step method in which the first step concerns a partitioning of the network by exploiting the covariance of links between nodes and the interclass inertia in order to generate small disconnected groups of nodes. The second step iteratively merges these initial groups obtained in the first step to identify the final community structure that maximizes the modularity.

A general approach to community detection consists in considering the network as a static view in which all the nodes and links in the network are kept unchanged throughout the study. Recent studies (Missaoui & Sarr, 2015) focus also on community evolution since most social networks tend to evolve over time through the insertion and removal of nodes and links. This leads to the growth or shrinking of groups inside the network and the displacement of members from one group to another one over time. Most of the studies on community

evolution use topological properties to identify the updated parts of the network and characterize the type of changes such as network shrinking, growing, splitting, and merging. However, recent work has focused on community evolution/detection by relying entirely on the behavior of group members in terms of the activities that occur in the network rather than exclusively considering links and network density.

Our research work is described in the next two chapters and brings new solutions to the field of network reorganization by taking inspiration from existing studies and filling the gap observed in predicting node substitution, link addition and cascading removal. We do not bring a new solution to the topic of community detection but we plan to exploit existing procedures as a preprocessing step of our solutions.

4. Network Restructuring After Node Disappearance

4.1 Introduction

In many real-life structures such as computer networks or organizations, each actor plays a more or less important role within that networks. Therefore, the disappearance of a *central* node such as a team leader or a mediator between groups or a *witness* (*i.e.*, the one who is the closest to the other nodes) will very likely not have the same impact on the network structure as the disappearance of a rather marginal one (*i.e.*, a node with a low centrality score). Hence, we need to find a substitute to any central node whenever it disappears. Such substitute needs to have a similar centrality score and role as the deleted node and should be linked to certain nodes in a predefined way so that the network structural properties are maintained, including its connectivity. To that end, we propose a role-based approach with its associated procedure to restructure a social network when a node disappears.

The objective of this chapter is to exploit network topology to devise an approach towards (i) finding a substitute to a deleted node if the latter is a central one, and (ii) adding appropriate links to maintain the network connected. The approach exploits the role played by nodes to predict the new structure of a social network once one entity disappears. The role of a node in the network is expressed in terms of its centrality in the network. Three important roles are considered: the *leader*, the *mediator* and the *witness*. An entity acts as a leader, a mediator or a witness if it has a high degree centrality, betweenness centrality and closeness centrality, respectively.

Many studies in social network analysis (SNA) focus on static networks as indicated in Jamali & Abolhassani (Jamali & Abolhassani, 2006). However, a social network is a dynamic structure where links and entities appear and disappear. In this chapter, we present a method to predict the new structure of a social network once a node is deleted. It consists on first finding a substitute to the deleted node whenever the latter plays a role in the network and then add new links to maintain the network connected. In practical terms, given a social network, what happens if an entity disappears? What will be the new structure of the network? Which node will become a substitute and play the role of the deleted node if the latter happens to be a leader or a mediator or a witness? What are the links that more likely will be created in order to maintain the network in a stable situation and a connected way?

The approach presented in this chapter is an extension to the study in Negre *et al.* (Negre et al., 2011) both in its structure and content:

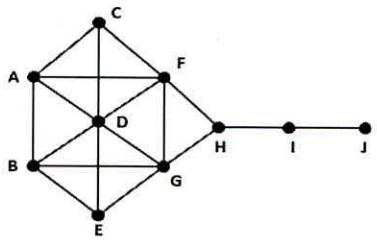
1. Algorithms are better presented and documented
2. The closeness centrality is introduced to define a new role called witness,
3. The specificity of the new network after its restructuring (node and link removal, substitute identification and link insertion) in terms of "return to normal state" relies on a richer set of network parameters such as density, number of nodes and edges, diameter and global clustering coefficient of the network.
4. Our approach is compared against Sarr's approach (Sarr & Missaoui, 2013) since the latter is the most related to our work.

In the following we present a method to predict the new structure of a social network once a node is deleted. It consists on first finding a substitute to the deleted node whenever the latter plays a role in the network and then add new links to maintain the network connected. In practical terms, given a social network $SN = \langle V, E \rangle$ where V is a set of N nodes (vertices), E is a set of interactions (edges) between vertices, we would like to predict what will happen if an entity $X \in V$ disappears and what the new structure of the network will be. Furthermore, we also would like to predict which node will become a substitute and play the role of the deleted node if the latter happens to be a leader or a mediator or a witness. Finally, new links are added based on a set of options that we have identified to maintain the network connected and stable.

4.2 Node Centrality and Link Options

The disappearance of central nodes (*i.e.,* the ones with at least one high centrality score such as degree centrality) will very likely not have the same impact on the network as the disappearance of rather peripheral ones (*i.e.,* nodes with low centrality scores). Based on this fact, we propose a procedure to predict the evolution of a social network structure after the deletion of one entity. Such a procedure first identifies the role of the removed node within the network, then effectively removes the node and its ties with its neighbors, determines the potential substitute and finally creates new links between the (unique) selected substitute (if it exists) and some identified nodes. Since degree, betweenness and closeness centralities are among the most important measures for the role or position of a node in a network, we will focus on them and associate to them three distinct roles, respectively: *Leader*, *Mediator* and *Witness*. When the deleted node has no role, its neighbors are then connected in a *clique* or a *chain* according to the needs of the user or the application.

Figure 9 provides three centrality measures for the nodes of the KITE network (Krebs, 2011).



Node	Degree Centrality	Betweenness Centrality	Closeness Centrality
A	0.444	0.023	0.529
B	0.444	0.023	0.529
C	0.333	0.000	0.500
D	**0.667**	0.102	0.600
E	0.333	0.000	0.500
F	0.556	0.231	**0.643**
G	0.556	0.231	**0.643**
H	0.333	**0.389**	0.600
I	0.222	0.222	0.429
J	0.111	0.000	0.310

Figure 9. Left: KITE network. Right: the centrality scores

When an individual disappears from the network (e.g., account closing, person's retirement), all the links associated with such a node are removed. Then, one substitute of the deleted node will be sought only when the vanished entity has played the role of a leader, mediator or witness in the network. The question is then the following: what is the impact of this disappearance on the remaining individuals? Will some individual form a clique for example? This will depend on the structure of the network, the role previously played by the deleted node in that network as well as the link option chosen by the user. We propose five possible options to link nodes in the network. We are aware that each option may be more appropriate than some other ones in some specific situations or real-life applications.

- *OLD*: A substitute node *i* is linked to the former neighbors of the deleted node *X*. *E.g.,* when Bob replaces the group leader Martin, then he initiates a new tie with anyone (not already linked to Bob) who was linked to Martin.

- *IMP*: A substitute node *i* is linked to a node *z* whenever the latter has a centrality (degree, betweenness or closeness) very close to (*i.e.,* deviating by at most a ratio ε from) that of *i*. *E.g.,* when Bob replaces the group leader Martin, he initiates a new tie with anyone who has almost the same role as his own role in the organization, *i.e.,* the one who has a degree centrality very close to his own centrality.

- *NGB*: A substitute node i is linked to node z if the latter has a number of common neighbors with i very close to the maximum number of common neighbors that i shares with other nodes. *E.g.,* if Jean and Daniel have many collaborators in common, then they will more likely collaborate (and link together) in the future.

- *CLIQUE*: this user-defined option may be chosen when no substitute is found or when the deleted node has no specific role. It aims at linking the former neighbors of the deleted node X in a clique. Such a situation may occur in real-life when a central family (or clan) member passes away and the other members decide to tightly connect to each other.

- *CHAIN*: this option is only used when no substitute is found or when the deleted node has no specific role. It aims to link the former neighbors of the deleted node X in a chain. In this case, neighbors are linked in a decreasing order of their degree centrality.

Using the network shown in Figure 9, let us remove node B which turns out to be a non-central node because none of its three centrality scores deviates for example by a ratio ε = 10% from the corresponding highest value of degree, betweenness or closeness centrality. The graph becomes the one in the middle of Figure 10 after the deletion of B. In such a case, all the nodes that were linked to B will form a clique (or a chain) as indicated in Figure 10-c.

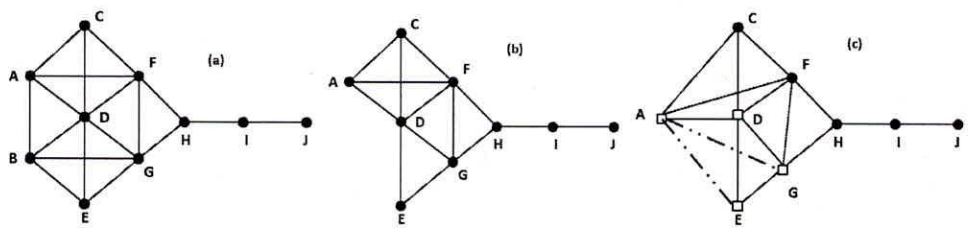


Figure 10. (a) The initial network. (b) After deleting node B and related ties. (c) After forming a clique with nodes A, D, E, and G that were neighbors of B

If however node D is deleted, a substitute is sought because the role of D is a *Leader* (see Figure 9 Right). If we tolerate a deviation ratio ε = 25% from the degree centrality of D, then a node with a degree centrality at least equal to 0.667 x 0.75 = 0.50 is a potential substitute. This is the case of nodes F and G whose degree centrality becomes 0.50 in the new network after removing D. Since F and G have the same score for the three centrality measures, one of

them is randomly taken. If F is the selected substitute for D, then the new network will be Figure 11-(b) if the link option is *IMP* or Figure 11-(c) if the link option is *OLD*. F is newly linked to B only in Figure 11-(b) because the centrality degree of B is equal to 0.375 and deviates at most by 25% from that of F (equal to 0.50). Nodes A and G are already linked to F.

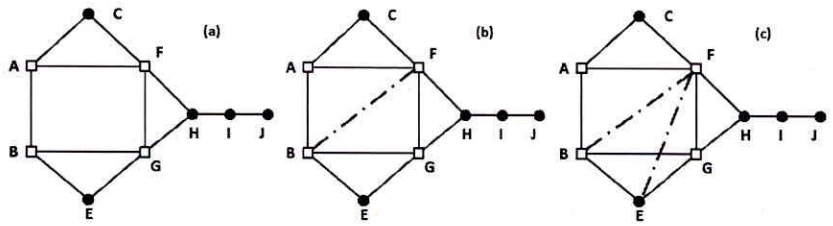


Figure 11. (a) The new network AFTER deleting node D and related ties. (b) AFTER using the link option *IMP* with $\varepsilon = 25\%$. (c) AFTER using the link option *OLD*

Using the normalized centrality formula (see Subsection 2.2), we get the following table which illustrate the importance of each node in the new network $\mathbf{SN}'$ after deleting D and then using the IMP link option with $\varepsilon = 25\%$.

Table 9. Centrality measures for the new network $\boldsymbol{SN'}$ after deleting D and using IMP to create new links

Node ID	Degree Centrality	Betweenneess Centrality	Closeness Centrality
A	0.375	0.071	0.500
B	0.375	0.071	0.500
C	0.250	0	0.440
E	0.250	0	0.440
F	0.500	0.286	0.610
G	0.500	0.286	0.610
H	0.375	0.429	0.610
I	0.250	0.250	0.440
J	0.125	0	0.320

(a)After the deletion of D

Node ID	Degree Centrality	Betweenneess Centrality	Closeness Centrality
A	0.375	0.029	0.500
B	0.500	0.095	0.530
C	0.250	0	0.440
E	0.250	0	0.440
F	0.625	0.360	0.666
G	0.500	0.190	0.610
H	0.375	0.428	0.615
I	0.250	0.250	0.440
J	0.125	0	0.320

(b)After using IMP with $\varepsilon = 25\%$

Another replacement alternative is to restrict the search for the substitute to the community of the deleted node because entities of the same community tend to share some common features. In our example, suppose that the set of nodes $\{A, B, C, D, E, F,$ and $G\}$ represents a community. Then, any search for the substitute of one of these nodes will be limited to its group. The benefit of such a restriction lies in the fact that the exploration of the network and the computation of centrality measures are limited to the community to which the deleted node belongs to rather than to the entire network. However, there are some real-life situations where it is much more valuable to explore the whole network to better find a substitute with a role close to that of the deleted node. For example, when a department chair leaves a company, we would like to find a good substitute not only inside the same community of the former head but throughout the whole organization.

Note that if the deleted node is a leaf (has only one link with the rest of the network), then there is no need to find a substitute or create new links between some existing nodes. In our example, the deletion of J from the initial network does not lead to the identification of a substitute or the insertion of new links.

4.3 Our Proposed Approach

The approach we describe in this subsection is based on the role played by nodes in a social network in terms of their interactions with other nodes to predict the new structure of that network after the deletion of one of its vertices. We recall that this method relies on the following hypotheses: (i) the network is a connected, undirected and unweighted graph, and (ii) two entities of the network are compared on the basis of their centrality measures in the network.

Let assume that node X disappears from the network. Then, we would like to identify the node of the network that will replace X and play a similar role as X, and then define the new structure of the network in terms of new and vanishing ties among nodes. Substitution is possible when X acts either as a leader, mediator or witness and when one node emerges as a substitute for the removed node.

4.3.1 Role

We define the role $r(SN, X)$ of a given node X within the network $SN = \langle V, E \rangle$ as an element of the finite set R = $\{Leader, Mediator, Witness\}$ such that

$$r(SN,X) \begin{cases} \text{Leader} & \text{If } \frac{C_D^{SN}(X)}{maxL} \geq 1-\varepsilon & \text{where } maxL \text{ is the maximal value of the observed degree centrality within } SN \text{ and } \varepsilon \text{ is a tolerance ratio.} \\ \text{Mediator} & \text{If } \frac{C_B^{SN}(X)}{maxM} \geq 1-\varepsilon & \text{where } maxM \text{ is the maximal value of the observed betweenness centrality.} \\ \text{Witness} & \text{If } \frac{C_C^{SN}(X)}{maxC} \geq 1-\varepsilon & \text{where } maxC \text{ is the maximal value of the observed closeness centrality.} \\ \text{NIL otherwise} & & \end{cases}$$

where $C_D^{SN}(X)$, $C_B^{SN}(X)$ and $C_C^{SN}(X)$ stand for degree, betweenness and closeness centrality measures respectively as defined in Subsection 2.2.

The value of ε is a user-defined ratio (*e.g.*, 10%) that indicates to what extent a given value is allowed to at most deviate from another one. When for example ε is null for role assignment, this means that the role of leader (or mediator/witness) is assigned to the deleted node if it has the highest degree (or betweenness or closeness) centrality in the network.

However, a non-null value of ε allows (*i.e.,* tolerates) a relative deviation of at most ε from the highest value of degree (or betweenness or closeness) centrality.

In case a node X has more than one role, it will take the role associated with its centrality that is most relatively close to the maximal corresponding centrality. For example, if node X has the roles of both *leader* and *mediator*, it will be a *leader* if $\frac{C_D^{SN}(X)}{maxL} \geq \frac{C_B^{SN}(X)}{maxM}$. Otherwise, X will be a *mediator*.

From Figure 9 Right, one can see that $maxL = 0.667$, $maxM = 0.389$ and $maxC = 0.643$. With a ratio $\varepsilon = 20\%$ of $maxL$, $maxM$ and $maxC$ respectively, the deleted node X has the role of a *leader* if it belongs to the set $\{D, F, G\}$, the role of a *mediator* if it corresponds to H, the role of a *witness* if it belongs to the set $\{A, B, D, F, G, H\}$, and has no role if it belongs to the set $\{C, E, I, J\}$. If the deleted node is F and $\varepsilon = 20\%$, F has the double role of *leader* and *witness*, then the computation of the two ratios $\frac{C_D^{SN}(X)}{maxL} = 0.8$ and $\frac{C_C^{SN}(X)}{maxC} = 1$ let us assign the role of witness to F.

Deleting a node from the network leads to deleting associated links. Based on the existing studies on link prediction and social network evolution, we believe that predicting the social network structure after the deletion of a given node cannot be limited to only the deletion (and possibly the insertion) of links but also to the identification of a node that can play a similar role as the deleted node.

With this observation in mind and assuming that the network should return to a "normal state" after a node deletion, our approach proceeds in two steps: (i) identify, if possible, the node of the network that will replace the deleted node X, and (ii) suggest a link option to tie some specific nodes. For instance, if a *mediator* is deleted, then a new candidate for mediation is sought. However, if the deleted node has no one of the three roles, then its removal will not have a great impact on the network and hence no substitute is required. However, new links will be added between the neighbors of the deleted node to keep the graph connected.

As indicated earlier, our approach takes into account the topology of the network to modify its structure after the deletion of one node.

- If the deleted node X has one of the three identified roles (*leader*, *mediator* or *witness*), then its links to other nodes are also deleted, but the past interaction between nodes via the deleted node should persist in some way.
- When the deleted node has a role and there is a substitute, then the latter will be linked to some other nodes of the network according to one of three established link options (*OLD, IMP, NGB*) as defined by Procedure *LinkS* (see explanation below).

- However, if no substitute exists for the central deleted node, then the neighbors of the deleted node are linked according to one of two possible link options (*CLIQUE, CHAIN*) using Procedure *LinkNoS* (see below). Then, the new network structure and the new central nodes are returned.
- If the deleted node has no role, *i.e.,* it is neither a leader nor a mediator/witness, then its links are also deleted but no substitute is sought because the node is not important enough to be replaced by another node. However, the nodes that were previously attached to the deleted one will form a clique or a chain.

Once the new network structure is established, if some nodes are completely isolated, *i.e.,* they are not linked to any other node, they are then deleted. This may occasionally occur when *IMP* and *NGB* link options are used to link the substitute to the neighbors of the deleted node.

4.3.2 Algorithms

In this section, we propose a main procedure (see Algorithm 1) called *PredictStruct* for predicting the social network structure after the deletion of one of its nodes. The algorithm incorporates the two steps of our approach: the identification of the substitute to the deleted node, and then the link management.

Given a social network *SN*, a deleted node *X*, and a ratio ε, the main procedure returns both the new predicted network *SN'* (where *X* and its related links are removed while new links are possibly added) and the new sets of leaders, mediators and witnesses of *SN'*. The ratio ε represents a tolerated relative deviation of a given value and is used as a parameter of some defined functions. For substitute identification (see *Substitute*(*X, SN,* ε*, Indic*)), ε shows how *Indic* (*e.g.,* degree centrality) value of a substitute could deviate from the indicator value of the deleted node. Using our illustrative example and ε = 25%, the potential substitutes for the leader *D* are those having a degree centrality at least equal to 0.667 x (1-0.25) = 0.50 in the network deprived from the deleted node and its associated links. This is the case of nodes *F* and *G* (see Table 9). The parameter ε used in Functions *LinkS* and *LinkNoS* represents the deviation of a measure (*e.g.,* the number of common neighbors) that a node *w* is allowed to have in order to be linked to *i*.

As indicated in Procedure *PredictStruct*, the input covers the initial network *SN*, the deleted node *X*, the user-defined tolerance ratio ε and the link options indicated in the vector *OpLink* of three elements. *Oplink*(*i*) option takes a value that depends on the value of *i*.

1. i = 1: use CLIQUE or CHAIN if there is no central node
2. i = 2: use CLIQUE or CHAIN if no substitute exists

3. i = 3: use OLD, IMP or NBG in case of a substitute.

```
1:  Procedure PredictStruct
2:  Input: SN = ⟨V, E⟩ : Initial network, X: deleted node, ε = tolerance ratio, OpLink: vector of link options
    Output: SN' (a global variable): new network; L, M and W: sets of leaders, mediators, and witnesses in SN'
3:  NB ← Neighbor (X); L, M, W ← ∅
4:  R ← Role(X, SN, ε)
5:  switch R do
      case Leader
            Indic ← 'D' ;
      case Mediator
            Indic ← 'B';
      case Witness
            Indic ← 'C';
      break;
      Otherwise
            SN' ← ⟨ V' = V – {X}, E'= E- {e_{x,yi} , ∀ y_i ∈ V'}⟩∪
            LinkNoS (SN', OpLink(1), NB)   /** CLIQUE or CHAIN **/
      end
    endsw
6:  if R ≠ NIL then
7:    NR ← Substitute (X, SN, ε, Indic)
8:    if NR ≠ NIL then
9:      SN' ← SN' ∪ LinkS (NR, SN', Indic, OpLink (3), ε, NB)
10:   else
11:     SN' ← SN' ∪ LinkNoS (SN', OpLink(2), NB)
12:   end if
13: end if
14: for i from 1 to (n-1) do   **Discard isolated nodes
15:   if ¬∃x ∈ V' such that e_{yi,x} ∈ E then /* e_{yi,x} is the edge between nodes x and y_i */
16:     V' ← V' - y_i
17:   end if
18: end for
19: L ← Leaders (SN')
20: M ← Mediators (SN')
21: W ← Witnesses (SN')
22: return (SN', L, M, W)
```

Algorithm 1. Predicting the structure of a network

After storing the number of neighbors *NB* of node *X*, the next executed instruction in the main procedure identifies the role of *X* (as defined in Subsection 4.3.1) via the function *Role*(*X, SN, ε*) which returns either *Leader, Mediator, Witness* or *NIL* (Line 4). The latter value indicates that *X* has no role. When X is a *leader* (*mediator* or *witness*), the parameter *Indic* takes the value 'D' ('B' or 'C' resp.) that stands for degree (betweenness or closeness resp.) centrality (Line 5). For each one of these three main roles, the procedure looks for a substitute to put in the set *NR* via Procedure *Substitute*(*X, SN, ε, Indic*) (Line 7 and Algorithm 2). If *NR* has a value that represents a node, this means that a substitute exists and hence will be linked to some other nodes of the network via the function *LinkS*(*NR, SN', OpLink, ε, NB*) as indicated in Line 9. Otherwise (*i.e.,* no substitute is found and NIL is returned), the procedure links the neighbors of *X* via the function *LinkNoS*(*SN', OpLink*(*i*), *ε, NB*) (Line 11).

1: **Procedure** *Substitute*
2: **Input:** $SN = \langle V, E \rangle$: Initial network, X: deleted node, ε = tolerance ratio, *Indic*: degree or betweenness or closeness centrality indicator.
Output: *Sub*: The substitute for X or the value NIL
3: $T_0 \leftarrow 0$
4: $ValX \leftarrow ValIndic(X, SN, Indic)$
5: $SN' \leftarrow \langle V' = V' - \{X\}, E' = E - \{ e_{yi,X} \ \forall \ y_i \in V \} \rangle$
6: **for** each $w \in V'$ **do**
7: $ValW \leftarrow ValIndic(w, SN', Indic)$
8: **if** $ValW \geq (1 - \varepsilon)$ x $ValX$ **then**
9: $T_0 \leftarrow T_0 \cup \{w\}$
10: **end if**
11: **end for**
12: $T_1 \leftarrow \arg\max_z \leftarrow (ValIndic(z, SN', Indic))$ such that $z \in T_0$
13: **if** $T_1 = 0$ **then**
14: $Sub \leftarrow NIL$
15: **else**
16: $Sub \leftarrow One\ (T_1)$
17: **end if**
18: **return**(*Sub*)

Algorithm 2. Finding the substitute of a deleted node

When the deleted node X has none of the three main roles, its neighbors in *NB* will form a clique or a chain as indicated in the *otherwise* part of the *switch* block (Line 5). Any isolated node is deleted (see Lines 14-18). Finally, the procedure computes the value of the three indicators for nodes of the predicted network SN' and returns SN' as well as the central nodes of the new network SN' (Lines 19-22).

Algorithm 2 describes the procedure *Substitute*(X, SN, ε, *Indic*). Given a social network SN, a deleted node X, a threshold ε and an indicator *Indic*, the procedure returns either a node as a substitute for X or the value *NIL* if no substitute is found. For that purpose, the procedure computes the value of the indicator *Indic* of the nodes in the network $SN' = \langle V', E' \rangle$, *i.e.,* without X and its associated links (Lines 4-5) and stores in T_0 the potential substitutes having an indicator value in SN' close to (*i.e.,* deviating at most by a relative proportion ε from) the indicator value of X in the initial network SN (Lines 6-11). If T_0 contains more than one substitute, only nodes with maximal indicator value are returned (Line 12) using the arg max function. If more than one argument has the maximal value, then one of them is randomly selected and returned as the substitute. If however, no argument exists, then the value *NIL* is returned (see Lines 13 – 18). The returned value of the substitute is stored in *NR* (see line 6 of Algorithm 1).

4.3.3 Additional Functions

Functions *Role, ValIndic, LinkS* and *LinkNoS* have the following meaning. *Role*(*i, SN,* ε) returns the role of a given node *i* within the network *SN* by looking for the maximal values *maxL, maxM* and *maxC* among the network nodes and by comparing them to the threshold ε (as defined in Subsection 4.3.1).

ValIndic(*i, SN, Indic*) returns the value of the indicator *Indic* (the degree, betweenness or closeness centrality) of the node *i* within the network *SN*.

LinkS(*i, SN', OpLink*(*i*), ε, *NB*) links the substitute node *i* in *SN* to other nodes in different manners depending on the chosen option *OpLink*(*i*). The considered options are: *IMP, OLD* and *NGB*. *IMP* links *i* to a node *z* in the network *SN'* when $C_{Indic}^{SN'}(z)$ deviates at most by a portion ε from $C_{Indic}^{SN'}(i)$ in *SN'*. *OLD* links *i* to nodes in *NB* (neighbors of *X*). *NGB* links *i* to node *z* if the latter has a number of common neighbors with *i* that deviates by at most a ratio ε of *p*, where *p* is the maximum number of common neighbors that *i* shares with other nodes in the new network *SN'*.

LinkNoS(*SN', OpLink*(*i*), *NB*) is used when no substitute is found and aims to create links between nodes in *NB* (neighbors of node *X*). The possible options in this case are *OLD* or *CHAIN*.

The time complexity of the two algorithms covers mainly the computation of centrality scores for all the *N* nodes in the network and a sorting of such scores $O(N\log N)$ to obtain the maximal values. It is then dominated by the complexity of computing the three centrality scores, mainly the betweenness and closeness centralities for all the *N* nodes in the network. We have seen in Section 2 that the time complexity for computing closeness and betweenness centrality measures are $O(NM + N^2\log N)$ and $O(NM)$ respectively where M is the number of edges. Therefore, the overall complexity of our algorithms is $O(NM + N^2\log N)$.

As an illustration, let us consider the network *SN* given in Figure 9 and apply Procedure *PredictStruct*(*SN, X,* ε, *OpLink*) with the following values: *X* = '*D*', ε = 25%, and *OpLink* = [*OLD, CHAIN, IMP*]. The node *D* to delete has the role of a leader since $\frac{C_D^{SN}(D)}{maxL} = \frac{0.667}{0.667} > 1 - 0.25$.. The variable *Indic* will then take the value 'D' (degree centrality). When Procedure *Substitute*('*D*', *SN*, 0.25, '*D*') is called, node *D* and its associated links are first deleted from *SN* as depicted in Figure 11-(a). Then, nodes *F* and *G* are potential substitutes because their degree centrality in the new network SN' (see the new scores in Table 9) is equal to 0.50 = 0.667 x (1 – 0.25). Let assume that *F* is selected as the substitute for *D*. When Procedure *LinkS*(*NR, SN', Indic, OpLinkI*(3), ε, *NB*) is called with the following values: *NR* = '*F*', *Indic* = '*D*', *OpLink*(3) = '*IMP*', ε = 0.25 and *NB* = {*A, B, C, E, F, G*}, we get the

network *SN'* displayed in Figure 11-(b) where *F* is newly linked to node *B* which is not already linked to *F* and is the unique node (other than F) in the network which has a degree centrality at least equal to (1 – 0.25) x 0.50. If however, *OpLink(3)* = '*OLD*', *F* is automatically linked to the nodes in *NB* that are not already linked to it (see Figure 11-(c)).

We believe that the proposed approach handles typical cases inspired from real-life situations. It can even work decently in some extreme situations like in complete networks where every node is linked to all the remaining nodes or in star graphs in which only one node is linked to the rest of nodes. For complete graphs with *N* nodes, the algorithm returns a new complete graph with (*N* - 1) nodes where the deleted node and its associated links are deleted. For star graphs, no substitute of the central node exists. In such a case, the link option consists in forming a clique or a chain with the remaining nodes depending on the need of the user or the application.

4.3.4 Community Detection

Looking for the substitutes within the whole network can be tedious and expensive. Let us consider for example a company where one of the leaders leaves (*e.g.,* retirement or firing). Then, the company will look for a substitute who has many interactions with individuals either within the community of the individual that left the company, or even outside his own community. Restricting the search for the substitutes inside a network community will reduce the computing time but assumes that a preprocessing of the network is conducted for community detection.

A possible improvement of our approach is then to first identify the communities within the network to further determine the role played by the deleted node *X* in its own community and find a substitute. If *X* plays a role of a leader or witness and there is a node (other than *X*) in its community which has a centrality score close to that of *X*, then it is selected. If no substitute exists, the neighbors of *X* are linked according to the previously defined function called *LinkNoS*. When the role of the deleted node *X* is a mediator, the scope of finding a substitute covers the set of communities in which *X* is involved.

4.4 Empirical Study

In this section, we conduct two series of tests. In the first series, we empirically evaluate the potential of our approach to predict the structure of the network when a node disappears. In the second series, we aim at comparing our approach against two methods, namely Sarr and Missaoui's method (Sarr & Missaoui, 2013) and the utilization of

Adamic/Adar measure (Adamic & Adar, 2003; Liben-Nowell & Kleinberg, 2003) to predict the new links to be added based on the proximity principle.

4.4.1 First Series

The first series of tests are carried out with three specific objectives in mind: execution time evaluation, substitute assessment, and network stability versus perturbation analysis.

To that end, we developed a prototype in Java on a Core 2 Duo E6750 with 2.66GHz and 3.23Go of RAM running under Windows XP and we used two data sets commonly known in the literature which represent large undirected and unweighted networks that we call COAUTHOR and POWER GRID[1], respectively. The data set *COAUTHOR* is a co-authorship network of scientists working on network theory and experiment, as established by Newman (M. E. J. Newman, 2006). It contains 1,589 nodes and 2,742 links. The data set *POWER GRID* represents the topology of the Western States Power Grid of the United States. The 4,941 entities are transformers, substations, etc while the 6,594 interactions are high-voltage transmission lines. Hence, the second data set is about three times larger than the first one but has a sparser structure than the co-authorship network.

Performance Analysis

The experiments conducted on the two data sets allow us to evaluate the time needed to predict the network structure after the deletion of one of its nodes. The two data sets have been evaluated in three cases:

1. No role of the deleted node or no existing substitute,
2. A given role (*Leader, Mediator* and *Witness*) and a substitute linked to the neighbors of the deleted node using the *OLD* option, and
3. A given role and a substitute linked to the neighbors using the *IMP* option.

The overall mean execution time for processing each one of the three cases five times is given in Figure 12. One may notice that the average time is generally small when the deleted node has no role. This is due to the fact that in such a case, there is no need to look for a substitute to be further linked in one of the three options (*OLD*, *IMP* or *NGB*) to the neighbors of the deleted node. Moreover, this time is slightly higher when option *IMP* is used rather than *OLD* to link the substitute to some other nodes in the network. This is due to the fact that in the former situation, the entire network will generally be explored to add links between the substitute and certain nodes while in the latter case, a generally small subset of the network

[1] These networks are available at http://www-personal.umich.edu/~mejn/netdata/.

(namely, the neighbors of the deleted node) is considered. The link option NGB is discarded due to the high computation time.

Finally, note that the execution time here is the CPU time needed to detect a possible substitute and add new links. The computation of centrality measures for each node is done outside our prototype using the software *UCINET (Chase & Borgatti)* .

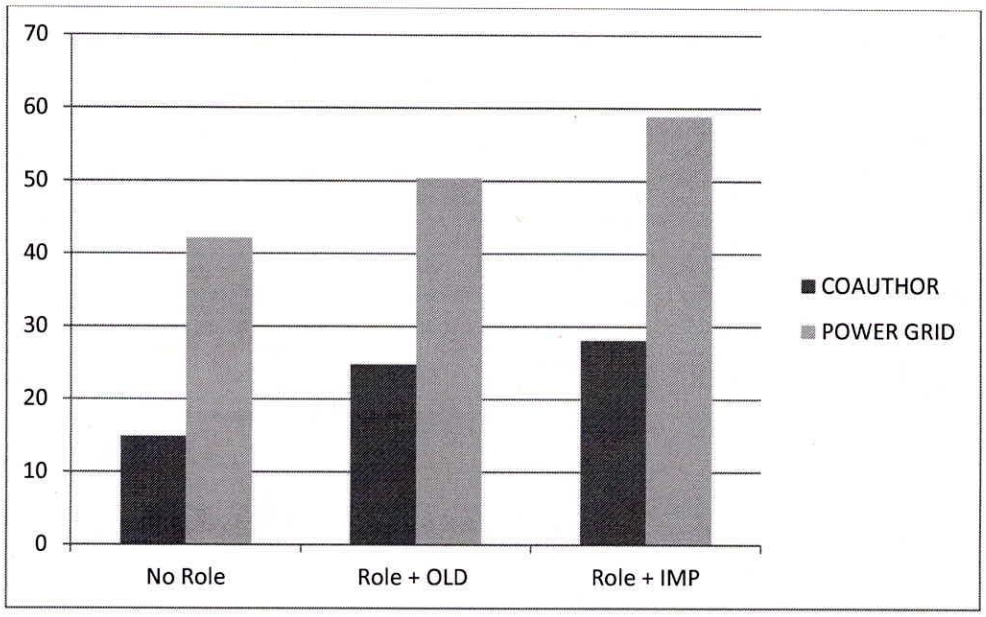


Figure 12. Execution time (in milliseconds)

As expected, Figure 12 shows that the execution time to predict the network structure after the deletion of one of its nodes increases with the network size.

Substitute Quality Analysis

To evaluate the quality of the prediction of a substitute and mainly the impact of ε on the identification of substitutes, we use the classical recall R and precision P measures as well as the F-measure labelled F, which are recalled as follows:

$$R = \frac{|S_{rel} \cap S_{retri}|}{|S_{rel}|}, P = \frac{|S_{rel} \cap S_{retri}|}{S_{retri}}, F = \frac{2 \times (R \times precision)}{(R + P)}$$

(Manning, Raghavan, & Schütze, 2009)

where S_{rel} is the set of relevant substitutes and S_{retri} is the set of substitutes retrieved by our main procedure with a tolerance ratio $0 \leq \varepsilon \leq 1$. The value S_{rel} is obtained by counting the number of nodes that have a centrality measure very close to that of the deleted node when

the latter has a specific role of leader, mediator or witness. This means that the tolerance ratio ε is almost zero, *i.e.*, we only consider the nodes with the highest centrality measures.

Figure 13 displays the recorded F-measure for the two data sets according to the value of 1 - ε. It shows that the curves have a classical appearance and that the smaller the ratio ε is (*i.e.*, the higher is 1 - ε), the better is the F-measure. For the two data sets, the F-measure is higher than 0.9 for $1 - \varepsilon \geq 75\%$, which is an indication that ε can reach up to 25% and still lead to a good substitute prediction.

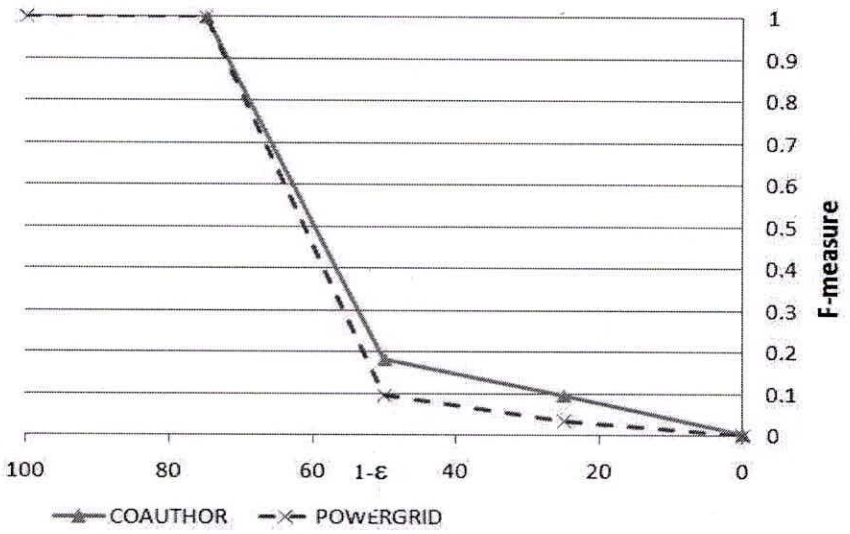


Figure 13. F-measure of the substitutes for the two data sets according to ε

Network Perturbation Analysis

The most popular (and easy to interpret) network measures for comparing two given graphs are single values (*e.g.*, clustering coefficient) rather than distributions (*e.g.*, the degree distribution) as stated by Kaiser (Kaiser, 2008). To analyze the perturbation of the network following a node removal and the conducted post-processing (substitute identification and link addition), we look at the mean variation (increase versus decrease) of five network measures before and after node deletion: density, diameter, global clustering coefficient as well as the number of nodes and links. In fact, a variation close to 0 is sought because it means an almost null perturbation and hence "a return to normal state" of the network following a node deletion.

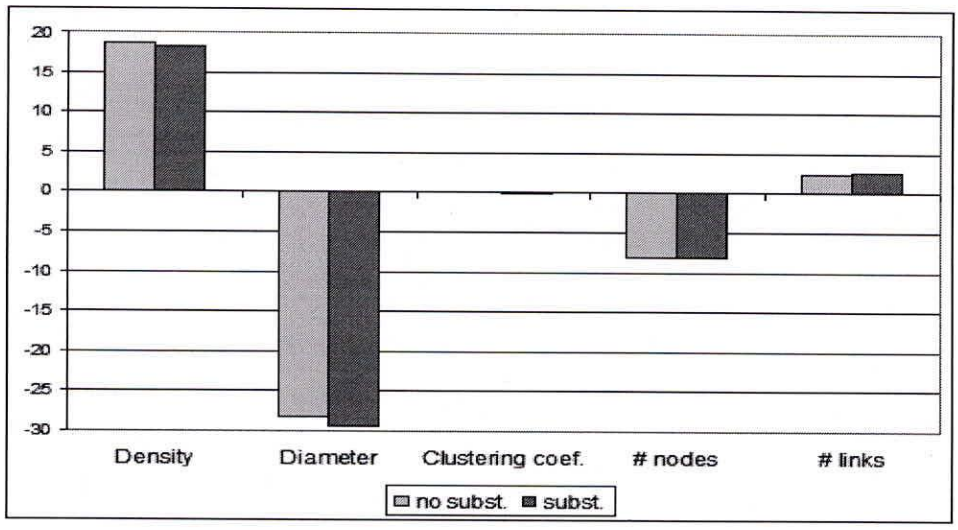


Figure 14. Network perturbation: COAUTHOR, ε = 20%, and with or without substitutes

Figures 14 and 15 exhibit the variation of network measures following a node removal for the data set COAUTHOR with ε = 20% and some variants. Figure 14 shows that the charts look similar in the two cases: presence versus absence of substitutes. In both cases, the clustering coefficient has almost a small variation. However, the density (resp. diameter) of the network has a relatively important increase of 18% (decrease of 27%) following node removal.

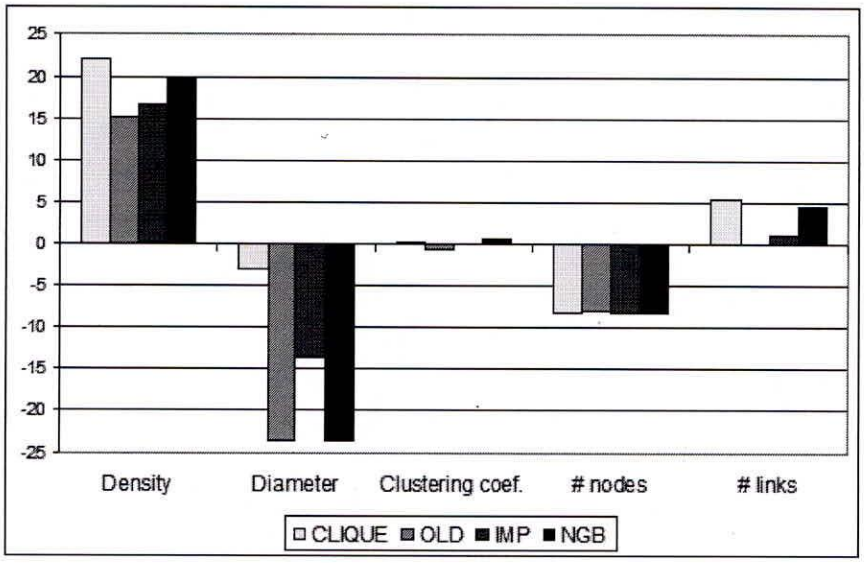


Figure 15. Network perturbation: COAUTHOR, ε = 20%, and link options

Figure 15 shows the impact of the link options (IMP, NGB, OLD and CLIQUE) on the network perturbation. The best results are often obtained for OLD and IMP because the variation of the overall network features is generally the smallest one (*i.e.,* no variation). However, NGB and CLIQUE options seem to be the worst ones because the network variations are the most important for the five measures.

We are aware that the conducted experiments are preliminary ones and that more extensive experiments are needed to validate our approach under various situations with varying network parameters (*e.g.,* size, density, clustering coefficient). This is the reason we conduct a second series of tests as described below.

4.4.2 Second Series

Our objective in the second series of tests is to compare our approach (called in the sequel UP for *User Preference*) against two recent studies that are the most comparable to it, namely the one described in Sarr and Missaoui (Sarr & Missaoui, 2013) and called IF for *Information Flow* and the usage of *Adamic/Adar* measure as a link prediction approach (labelled A/A) described in (Adamic & Adar, 2003) and found to be effective for link prediction in (Liben-Nowell & Kleinberg, 2003).

Adamic/Adar's method links the nodes that present some similarity. To predict whether a node has a tie to other ones, all nodes are ranked based on their similarity to this specific node. $similarity\ (i,j) = \sum_{shared\ items} \frac{1}{log[frequency\ (shared\ items)]}$. For instance, if two nodes have many friends in common, these two nodes are likely to become friends in the future. We adopt the principle of *Adamic/Adar* in the following way:

- Given a social network $SN = \langle V, E \rangle$, consider all edges $e_{ij} \in E$ with a shortest path at least equal to two *(e.g.,* all pairs of non-adjacent nodes with at least one common neighbor).
- The score of each pair of nodes (i, j), as explained in Liben-Nowell and Kleinberg (Liben-Nowell & Kleinberg, 2003) is computed as follows: $core(i,j) = \sum_{z \in \Gamma(i) \cap \Gamma(j)} \frac{1}{\log |\Gamma(z)|}$ where $\Gamma(i)$ is the set of neighbors of node i, $\Gamma(j)$ is the set of neighbors of node j and $\Gamma(z)$ is the set of common neighbors to both i and j.
- Choose the highest score of edge pairs of nodes from the step above. We called it $max_{a/a}$.
- Set the threshold ε and a lower bound (lb), where $lb = max_{a/a} - (\varepsilon * max_{a/a})$
- Link all pairs of nodes (i,j) with a *score* $(i,j) > lb$.

- The last step of this link prediction method is to discard the node concerned by the node removal.

In this second series of our tests, we simplified our own approach in order to make the comparison more significant and coherent with Sarr and Missaoui's approach (Sarr & Missaoui, 2013)

- The results from centrality measures (degree, betweenness and closeness centrality) are being sorted in a descending order. Moreover, ε is set to zero in order to consider only the nodes with the highest centrality values.
- Since in Sarr and Missaoui's approach (Sarr & Missaoui, 2013) the focus is on maintaining the information flow quality, when a node has two roles, the witness role is preferred to the leader position which is preferred to the mediator role.
- Only the option *CHAIN* is used when the deleted node X has no role or has a role but no substitute. It is exploited to link the neighbors of X in a descending order of their centrality value.
- When node X has a role, a node S will act as a substitute to the deleted node X if its centrality measure is equal to the corresponding one for node X. Otherwise, no substitute is suggested.
- When a substitute exists, *OLD* or *IMP* is used to link the substitute to the neighbors of the deleted node. *NGB* option has been discarded because it is very time-consuming.

We recall that the objectives in Sarr and Missaoui (Sarr & Missaoui, 2013) are (i) to avoid the degradation of the *information flow* due to the removal of one or many nodes, and (ii) to identify the substitute of the deleted node(s) based on its (their) role within the network.

Let $\gamma_F(SN) = \langle \lambda_F(SN), \varepsilon_{SN}(w)\rangle$represent the information flow quality within a social network SN where $\lambda_F(SN)$ denotes the *Information Flow* degree and $\varepsilon_{SN}(w)$ is the eccentricity of the witness in the network, i.e., the node with the smallest eccentricity value. We consider a social network $SN' = \langle V', E'\rangle$, where V' is a set of nodes without the deleted nodes and E' is a set of edges without the links with the deleted nodes. The algorithms in Sarr and Missaoui (Sarr & Missaoui, 2013) work as follows:

- The nodes are divided into two categories; (i) critical, and (ii) non critical ones. Critical nodes are those with the highest centrality measure, while non-critical nodes are the rest of the nodes. A non-critical node is either a *finger* or a *follower*. A *finger* is a node whose centrality deviates slightly from the critical node. For instance, let $C_D(L)$ be the degree centrality of a leader, a node i with degree centrality $C_D(i)$ is

considered as a finger of a leader if $\frac{C_D(L)-C_D(i)}{C_D(L)} \leq \varepsilon$ where ε is a tolerated relative deviation. A node is considered a *follower* if its degree centrality relatively deviates from the highest $C_D(L)$ by more than ε.

- A node has a role if it is critical. There are three possibilities to find a substitute when a critical node (*Nc*) is removed from the network:
    - There is at least one finger associated with *Nc*. The first finger F_{Nc}^1(*i.e.*, the one with the highest centrality value) is selected as a substitute.
    - When no finger of *Nc* exists, a critical node with the shortest geodesic distance to *Nc* is used as a substitute.
    - Otherwise, the neighbor of *Nc* with the highest centrality measure is selected as a substitute.
- There are two cases to consider when removing non critical nodes from the network *SN*: (i) the new network *SN'* is connected, and (ii) *SN'* is disconnected. When *SN'* is connected, no action is taken since the *information flow* will not be affected by the node deletion. However, when *SN'* is disconnected:
    - A direct link is added between neighbor of deleted node (*j* or *k*) if one of them is critical
    - An indirect link is added between *j* and *k* through a node *s* if only one of the two nodes *j* or *k* is critical and *s* has the shortest path to that critical node.

The three main differences between our approach and Sarr's approach (Sarr & Missaoui, 2013), as indicated in Subsection 4.4.2, is as follows: (i) we are concerned with the stability of the network structure while the latter focuses on the quality of information flow, (ii) the link options are chosen by the user in our case while in the latter approach nodes are linked in a parsimonious way according to a predefined procedure that gives priority to central nodes and mainly to the witness, and (iii) a role (*e.g.*, a leader) is associated with a deleted node in our approach if the corresponding centrality measure (*e.g.*, degree centrality) of the node deviates slightly from the highest score while in Sarr *et al.* a role is assigned to a node only if it has the highest score. The last observation means that in our approach, the central node is either a critical one or a finger as defined in Sarr's method.

In the following, we use centrality measures (*e.g.*, degree, betweenness and closeness) and network measures (*e.g.*, lowest eccentricity, density, diameter, and clustering coefficient) to conduct a comparison of the three methods discussed above. For the second series and in order to be compatible with the implementation of Sarr's approach, we used a single machine Intel Core i5 (8 GB of RAM and 3.20 GHz in Linux Ubuntu Platform) and coded our method using Python programming language within the network library *NetworkX* framework.

In this group of tests, we used three networks, namely the *karate club, collaboration and Barabási* networks.

1. The *karate club network* (M. E. J. Newman, 2013) is an undirected and unweighted network of a karate club at a US university in the 1970s. The network consists of 34 members and 78 relationships among these members. The density of the network is 6%.
2. The *collaboration network* is a large and sparse one that represents the collaboration network of authors whose papers were submitted to High Energy Physics, Theory category. The data cover papers from January 1993 to April 2003, as described in Leskovec *et al.* (Leskovec, Kleinberg, & Faloutsos, 2007) and can be retrieved at Stanford dataset[2]. From the original network, nearly 2,000 nodes are randomly removed. The nodes that do not belong to the biggest component are removed to get one giant network that represents our experimental data set. The final network contains 5,939 nodes and 18,708 edges with a density of 0.1 %.
3. The *Barabási network* is a large and dense network. It represents a scale-free network obtained through the Barabási model, *i.e.,* a network whose degree distribution follows a Power law (Barabási & Albert, 1999). First, a network with 2,000 nodes and a probability of 0.32 is generated. Then, we delete the most central nodes in order to have different nodes for each role. The final network contains 1,996 nodes and 634,373 edges with a density of 32 %.

The comparisons are done for critical versus non critical nodes, sparse versus dense networks, *OLD* vs *IMP* link options, and for *User's Preferences* (UP), *Adamic/Adar's* (A/A) and *Information Flow* (IF) approaches. The x-axis represents the compared approaches while the y-axis expresses the variation of edges, processing time, centrality and topology measures. The test results are shown in the figures below.

[2] The data can be retrieved from http://snap.stanford.edu/data/index.html

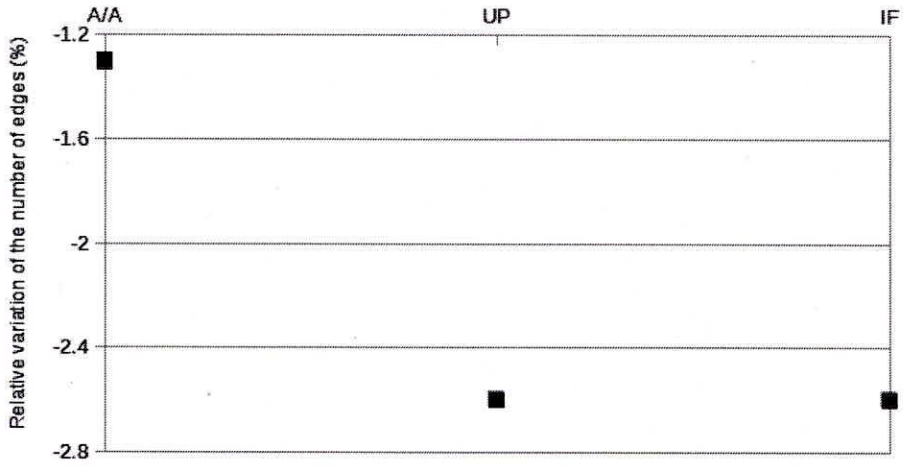


Figure 16. Variation of the number of edges for the *karate club network* with a non-critical node removal

Figure 16 shows that for a non-critical node removal in the karate club network, the number of edges decreases for UP and IF but very slightly for A/A. Using either OLD or IMP option for UP (see Figure 17), we note a larger decrease of the link number for UP with IMP option, IF and A/A, while the OLD option leads to a smaller decrease. A/A has a decrease of a same order as IF.

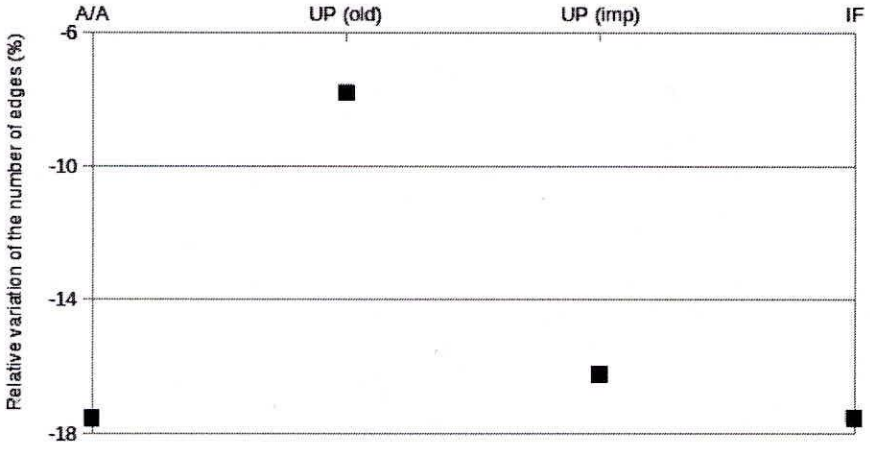


Figure 17. Variation of link number for the *karate club network* with a critical node removal

One can see that the processing time for both types of (critical and non-critical) node removal is the smallest one for UP (see Figure 18 and 19) in the karate club network.

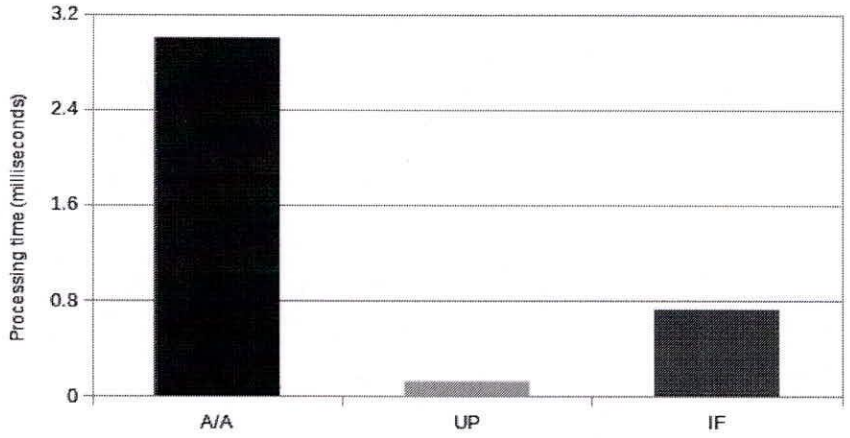


Figure 18. Processing time for the *karate club network* with a non-critical node removal

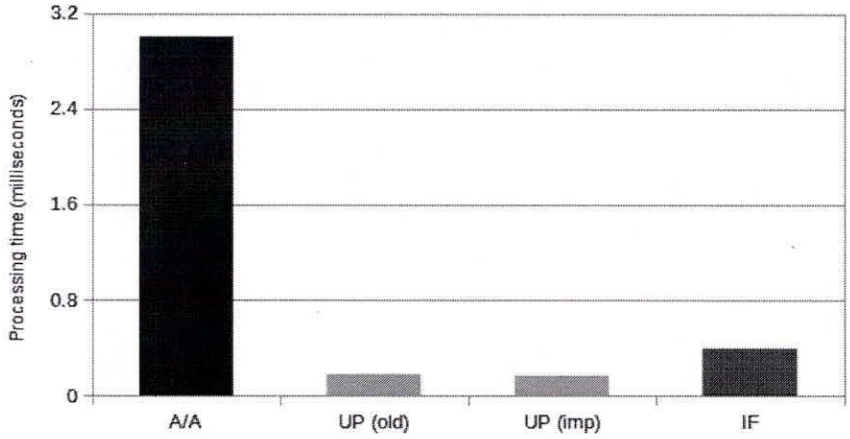


Figure 19. Processing time for the *karate club network* with a critical node removal

Figure 20 shows that in case of non-critical node removal the A/A approach leads to an increase in the degree and closeness centrality measures and a decrease in the betweenness centrality score while UP and IF have a similar behavior.

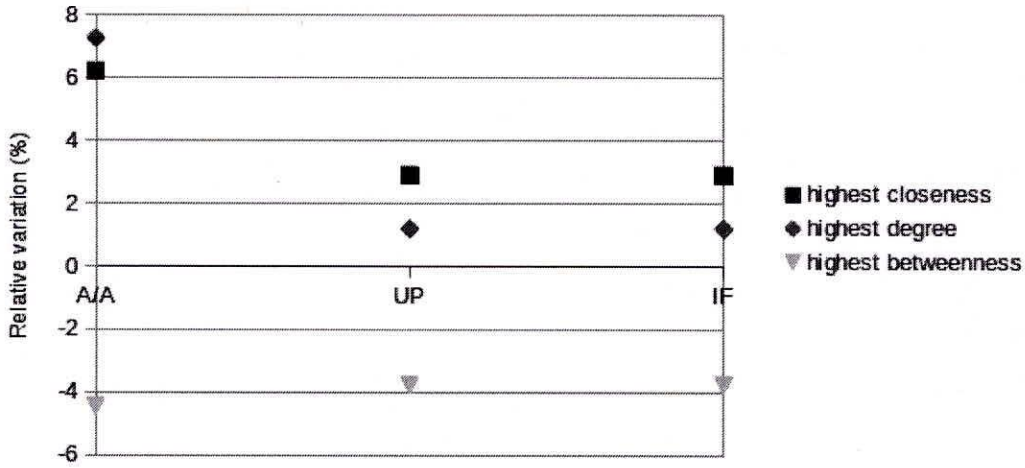


Figure 20. Variation of centrality measures for the *karate club network* with a non-critical node removal

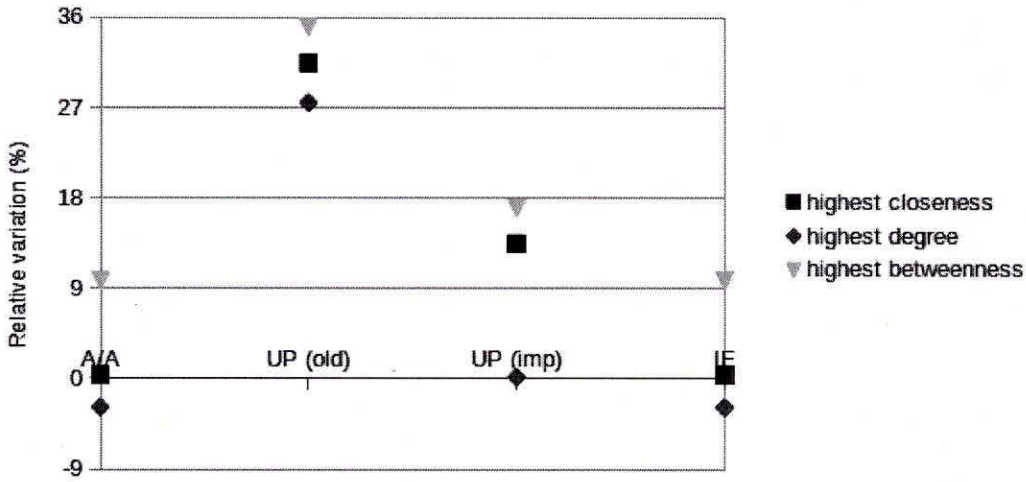


Figure 21. Variation of centrality measures for the *karate club network* with a critical node removal

Figure 21 shows that A/A and IF have a similar behavior with respect to the centrality measures. One can also note that in these two cases, the degree centrality slightly decreases while it does not change with the IMP link option of UP. The OLD link option of UP gives the highest variation of the centrality scores. We can then suspect that the usage of this option artificially increases the importance of nodes.

From Figure 22, we can see that UP and IF approaches have similar values for the topological measures. However, the density and clustering coefficient of the A/A approach are higher compared to UP and IF.

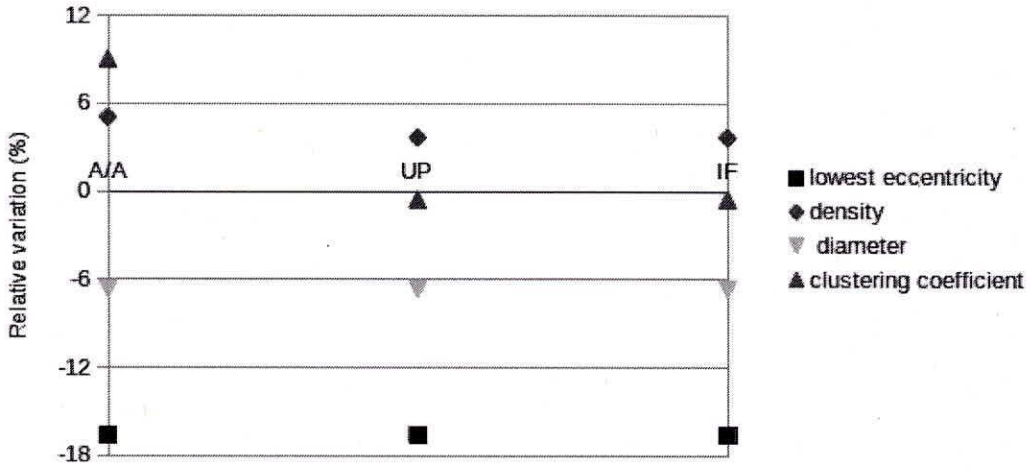


Figure 22. Variation of topological measures for the *karate club network* with a non-critical node removal

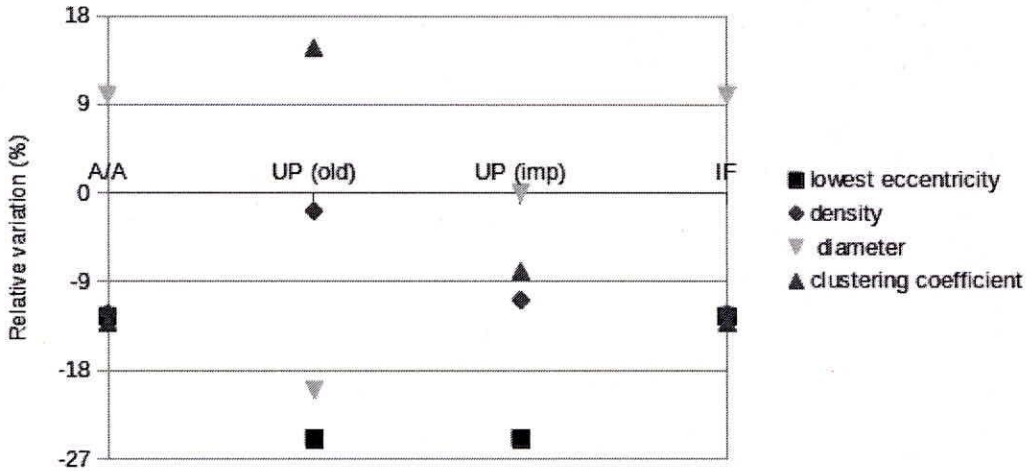


Figure 23. Variation of topological measures for the *karate club network* with a critical node removal

Figure 23 shows that in case of a critical node removal, IF and A/A have a similar behavior *w.r.t.* the variation of topological measures. The IMP and OLD options of UP do not have a

similar impact on the network structure except for the eccentricity variation. The IMP option is then preferred to the OLD option because it leads to a more stable structure. Since the IF approach focuses on maintaining a good quality of information flow, the decrease of eccentricity and the clustering coefficient is relatively small. However, there is a slight increase in the network diameter.

Table 10 summarizes our findings and comments for the karate club network.

Table 10. Comparison between non critical versus critical node removal for the *karate club network*

	Non critical node removal	**Critical node removal**
Number of edges	Fewer edges compared to the original network. UP and IF are comparable.	The same like non-critical nodes. OLD option of UP performs better.
Processing time	A/A is the slowest one while UP is the fastest one.	Same like non-critical node removal
Centrality measures	A/A gives the worst results while UP and IF give almost the same centrality measure values.	A/A and IF have similar centrality measure variations. IMP gives better results than OLD while IF gives good results compared to UP
Topological measures	UP and IF give similar results. The lowest eccentricity is similar for the three approaches	Overall, UP is more stable compared to A/A and IF

The question now is as follows: do we come to similar conclusions interms of network reorganization and processing time for larger networks and mainly the dense ones? The answer will be given based on the two other networks, namely *collaboration* and *Barabási* networks.

We recall that for a non-critical node removal, only CHAIN link option is used with the UP approach. From Figure 24, one can see that the variation of the link number for UP and IF is almost the same. The variation of the number of edges for A/A is however higher than for UP and IF.

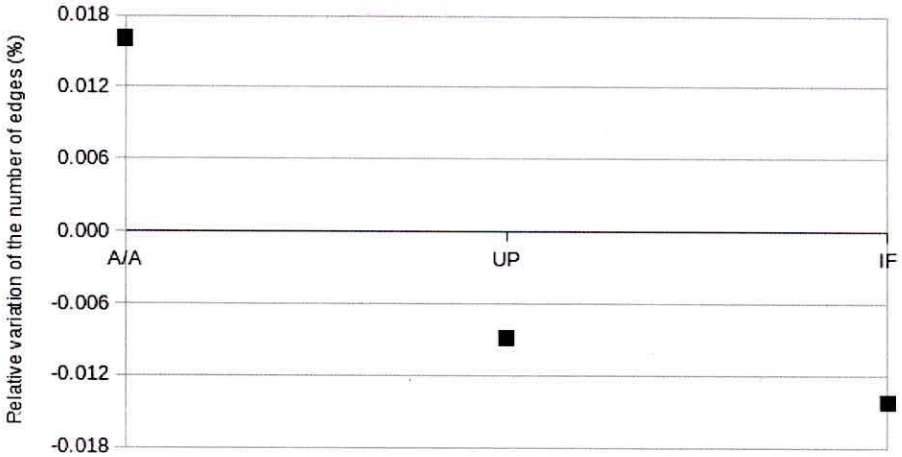


Figure 24. Variation of link number for *collaboration network* with a non-critical node removal

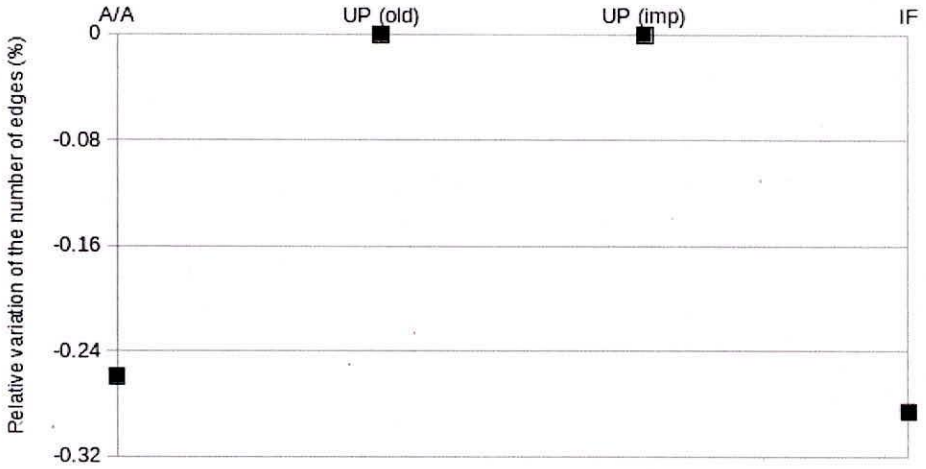


Figure 25. Variation of link number for *collaboration network* with a critical node removal

As shown in Figure 25, the insertion of new links in a parsimonious way in the IF approach leads to a decrease of the number of links following a critical node removal. UP gives no variation of the number of links for the two link options, which is a good sign for a return to a stable situation.

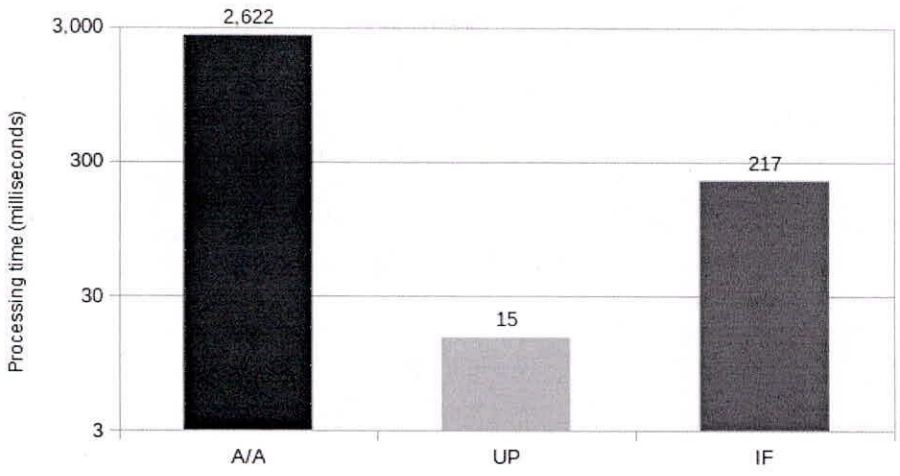


Figure 26. Processing time for *collaboration network* with a non-critical node removal

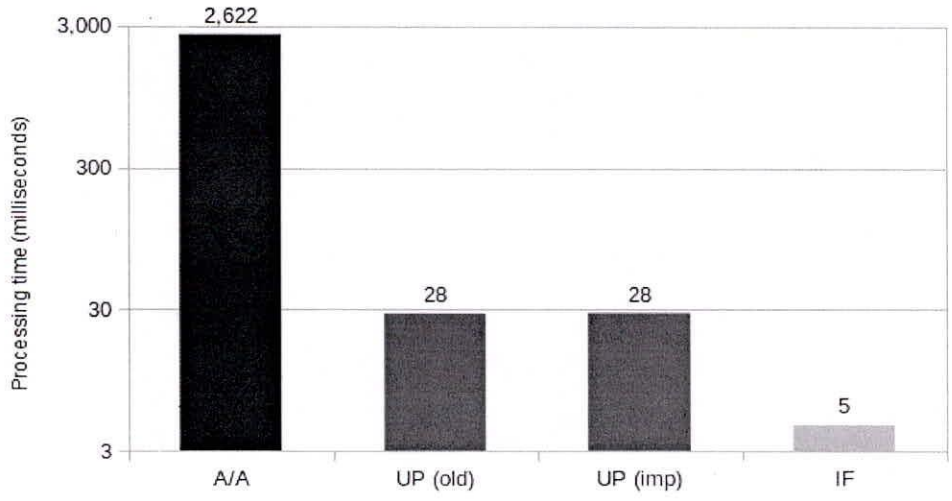


Figure 27. Processing time for *collaboration network* with a critical node removal

The high processing time for A/A in Figures 26 and 27 comes from the need for common neighbor computation which is known to be very time-consuming.

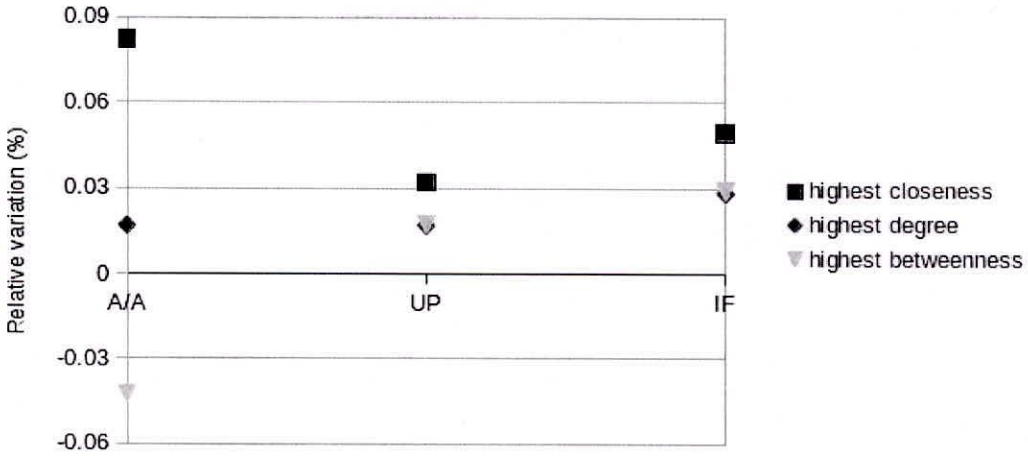


Figure 28. Variation of centrality measures for *collaboration network* with a non-critical node removal

The closeness, degree and betweenness centrality scores of the UP approach are almost similar to that of IF since they both increase very slightly. The increase of closeness centrality for IF validates the approach whose main objective is to maintain the information flow quality at a good level. The high closeness value in the A/A approach may be due to the proximity principle behind A/A.

From Figure 29, one can see that after a critical node removal in the *collaboration network*, the centrality value variations of A/A and IF are very similar while in Figure 30 the OLD and IMP link options have similar variations.

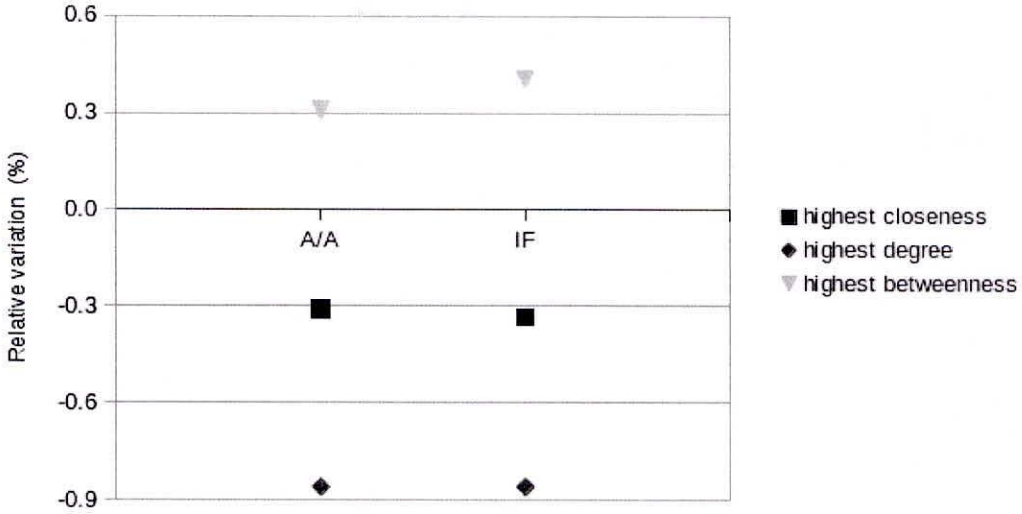


Figure 29. Variation of centrality measures in case of A/A or IF for *collaboration network* following a critical node removal

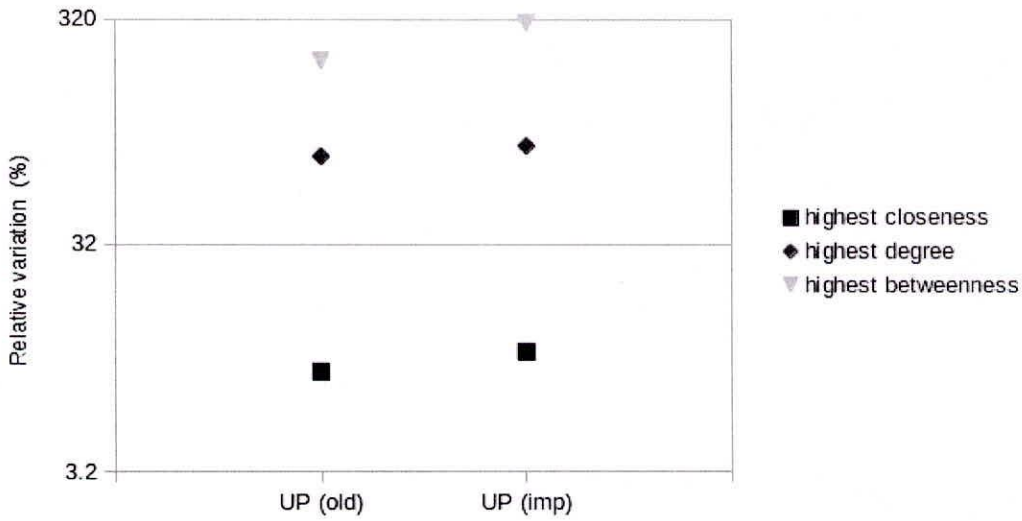


Figure 30. Variation of centrality measures using OLD or IMP for *collaboration network* after a critical node removal

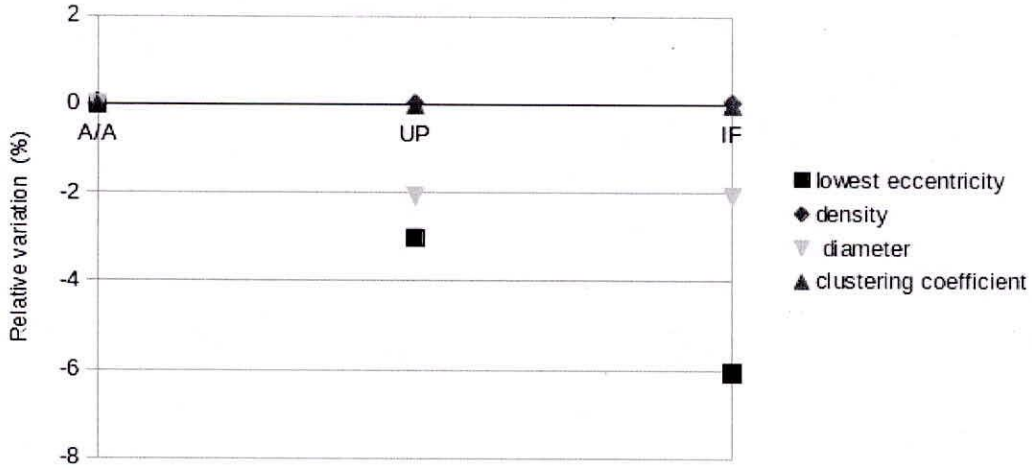


Figure 31. Variation of topological measures for *collaboration network* with a non-critical node removal

For the variation of topological measures in case of a non-critical node removal, the A/A approach leads to a more stable structure. The low value of eccentricity in IF is normal since IF aims at maintaining information flow quality.

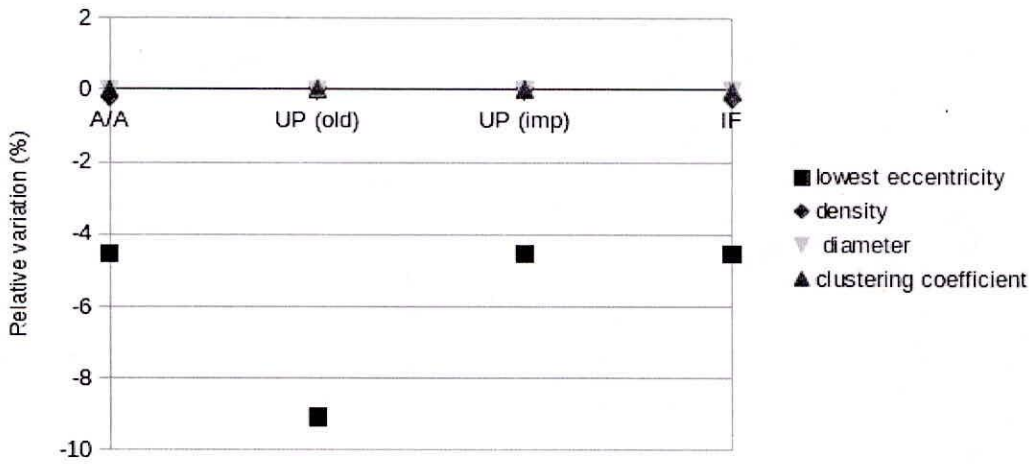


Figure 32. Variation of topological measures for *collaboration network* with a critical node removal

Despite the relatively high decrease of eccentricity values, the A/A, UP and IF approaches have a similar behavior *w.r.t.* to topological measures after a critical node removal (see Figure 32).

The following table summarizes our observations.

Table 11. Non critical versus critical node removal for *collaboration network*

	Non critical node removal	**Critical node removal**
Number of edges	A/A leads to more edges compared to the original network while UP and IF have fewer.	UP has the same number of edges as the original network while A/A and IF have fewer.
Processing time	A/A has the highest processing time followed by IF.	IF has the fastest processing time while A/A has the highest one.
Centrality measures	Overall, UP gives better results compared to A/A and IF.	A/A and IF give better results compared to UP.
Topological measures	A/A leads to a more stable structure compared to the other two methods.	The lowest eccentricity value has decreased for A/A, UP and IF.

In the following we give details about using the *Barabási* dense network.



Figure 33. Variation of link number for the *Barabási network* with a non-critical node removal

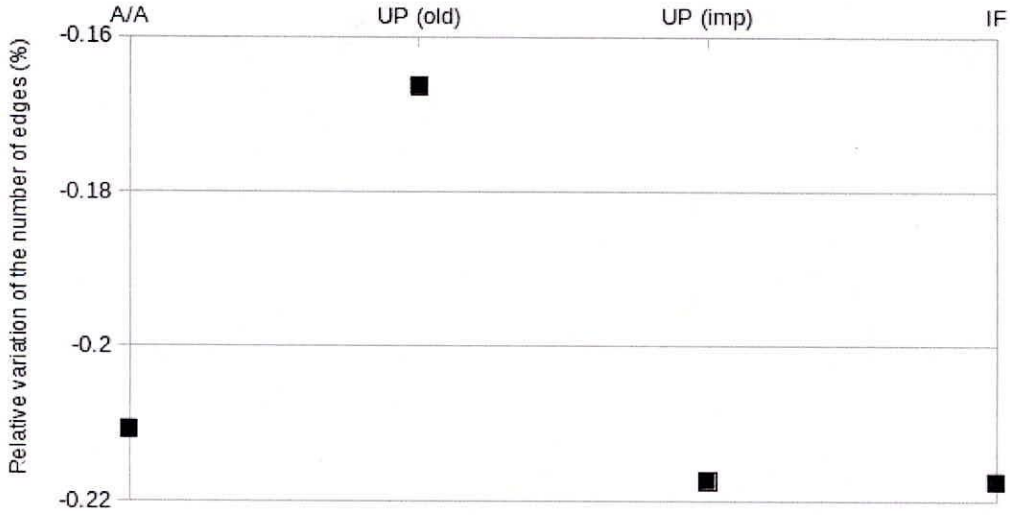


Figure 34. Variation of the link number for the *Barabási network* with a critical node removal

Figures 33 and 34 show that the variation of the number of links is the smallest for the OLD option when we are dealing with a large and dense network.

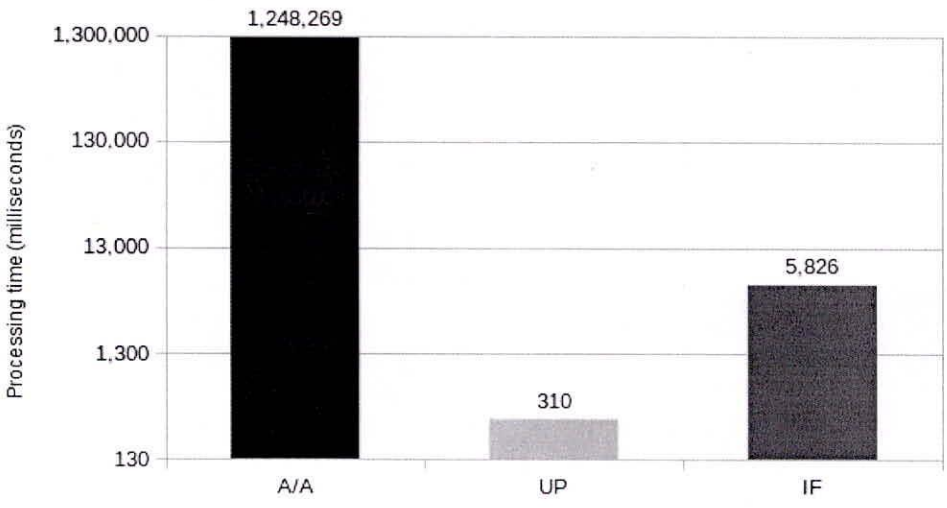


Figure 35. Processing time for the *Barabási network* with a non-critical node removal

Using the Barabási network, the UP approach has the lowest execution time after a non-critical node removal while in the critical node deletion, IF performs better.

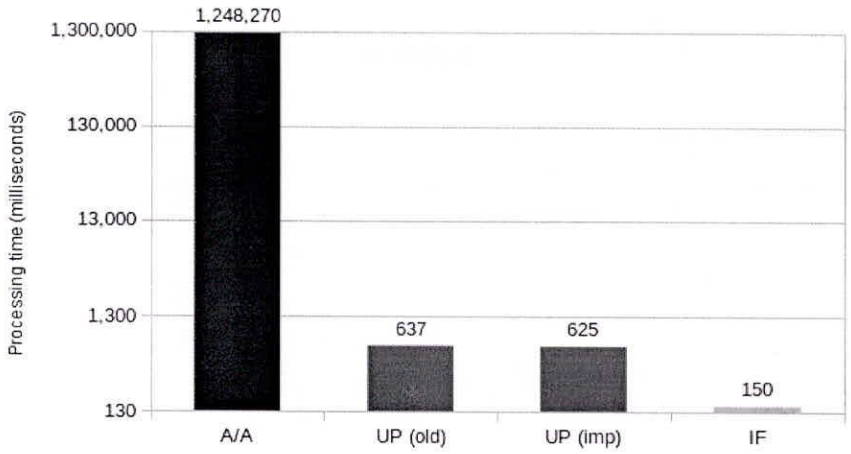


Figure 36. Processing time for *Barabási network* with a critical node removal

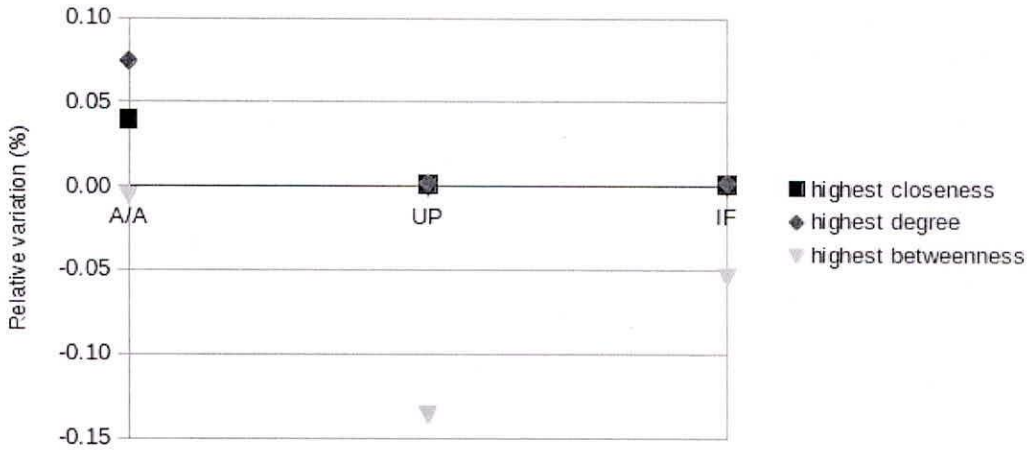


Figure 37. Variation of centrality measures for Barabási network with a non-critical node removal

As shown in Figure 37 and following a non-critical node deletion, centrality values are more stable (almost a null variation) *w.r.t.* to degree and closeness scores for UP and IF while a slight increase of such scores is observed for A/A. However, Figure 38 shows that in case of a critical node removal, the A/A approach exhibits more stability, *i.e.,* the least variation, followed by the IMP option of UP and then IF.

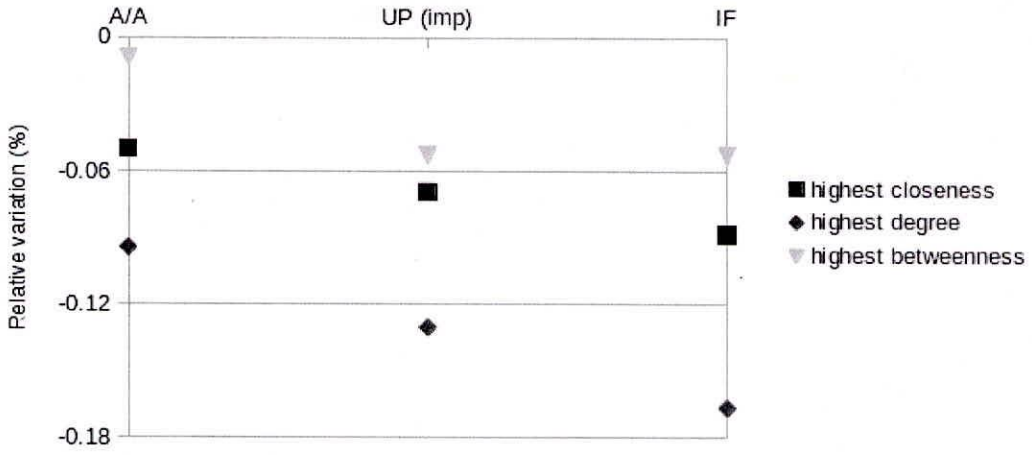


Figure 38. Variation of centrality measures for the *Barabási network* with a critical node removal

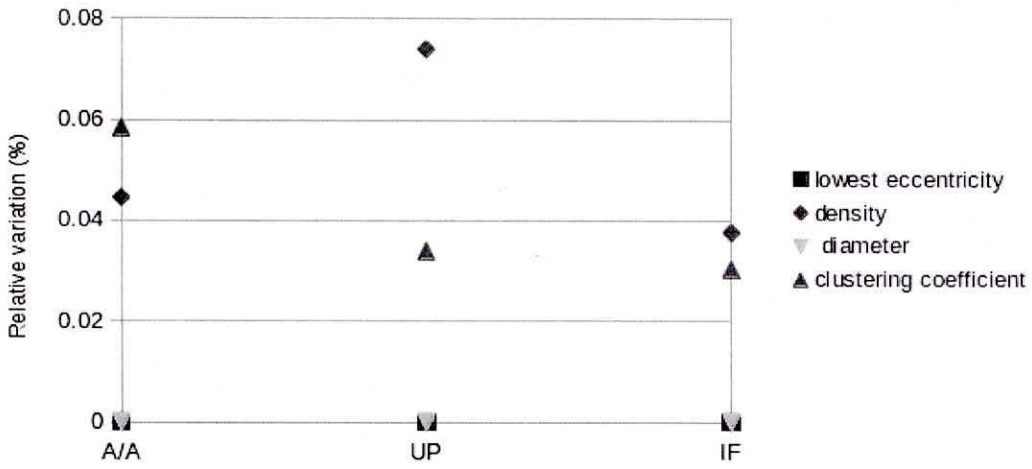


Figure 39. Variation of topological measures for *Barabási network* with a non-critical node removal

The variation of topological measures following a non-critical node removal is similar *w.r.t.* eccentricity and the network diameter for the three methods. The density increases slightly for all the methods and mainly for UP. The clustering coefficient variation is the same for UP and IF but very slightly higher for A/A.

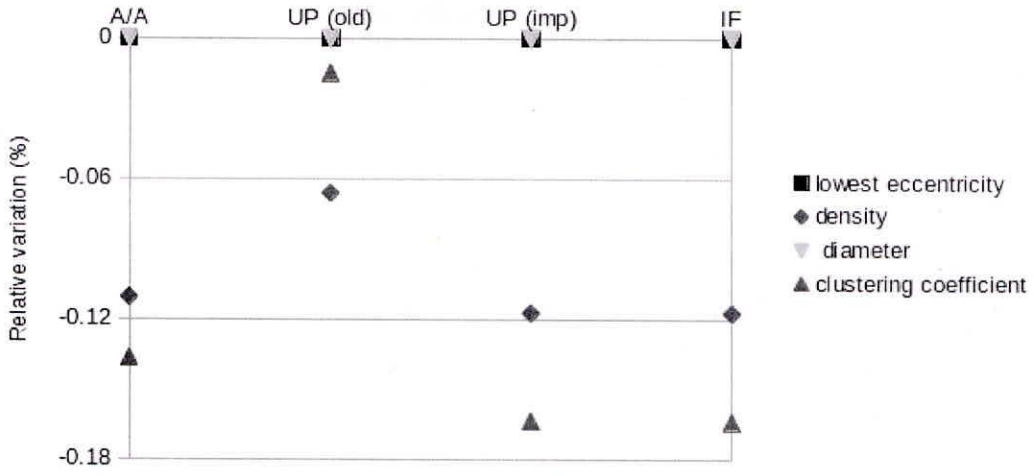


Figure 40. Variation of topological measures for the *Barabási network* with a critical node removal

Figure 40 shows no change for eccentricity and density in the four cases following a critical node removal. Both IF and the link option IMP of UP have the same behavior. Like for the *collaboration network*, the OLD option of UP provides the smallest variation and hence leads to better network stability.

Table 12. Non-critical versus critical node removal for the *Barabási network*

	Non critical node removal	Critical node removal
Number of edges	UP gives better results compared to A/A and IF.	The OLD option of UP leads to the smallest variation compared to A/A, IF and the IMP option of UP.
Processing time	UP is the fastest one while A/A computation time is the highest.	The processing time of IF is the best one compared to UP (both OLD and IMP) and A/A
Centrality measures	IF has the smallest variation	A/A has the smallest variation followed by the IMP option of UP
Topological measures	IF has the smallest variation and hence better network stability	The OLD option of UP has the smallest variation.

While we set the A/A approach as a link prediction method based on the proximity of nodes using the Adamic/Adar formula, the UP and IF approaches add new links based both on node centrality and proximity. The computation time in A/A is the worst both for sparse and dense networks and following either a critical or a non-critical node removal. Since none of the three approaches is the best in all situations either in terms of processing time and network structure variations in the two cases of node removal, we believe that depending on

the size and the density of the analyzed networks on one hand, and the type of removal (critical versus non-critical) on the other hand on, one may favor a given method rather than the other ones in some specific situations.

4.5 Conclusion

In this chapter, we have proposed a role-based approach for social network structure prediction once a node is deleted from the network. This approach, inspired from human behavior in social and professional situations and an enriched version of the study in Negre *et al.* (Negre et al., 2011), is a first step toward a prediction method of social network structure when one entity disappears and a substitute is needed. It first selects a substitute to the disappearing node whenever the latter is central in terms of its centrality measures, and then adds appropriate links based on the user's preferences. Three types of central nodes are considered based on degree, betweenness and closeness centrality scores respectively: *Leader*, *Mediator* and *Witness*.

We proposed link options such as OLD, IMP, NGB, CLIQUE and CHAIN by taking inspiration from real-life situations and known references. To illustrate the NGB (common neighbors) option for example, we can state that if an individual X is sharing common friends with another person Y in Link, it is very likely that X and Y will become connected in the future.

Our experiments conducted on known social networks and using two other similar approaches show that our approach has nice features. It can reorganize a social network structure in a reasonable time while link options OLD and IMP offer relatively good results to maintain the network in a stable situation.

We will explore additional link options since a given set of nodes can be tied in different ways. We plan also to analyze the impact of first identifying and exploiting communities on network reorganization following a node disappearance and the search for a substitute.

More importantly, an interesting extension of this work is to consider the removal of links rather than nodes and propose a new way to reorganize the network. For example, in the *Route network* used in the empirical study, it is more realistic to observe the deletion of one or many connections between two given airports than the deletion of an airport. In such a case, it is interesting to forecast the impact of route removal on other routes used by airline companies and flights.

5. Cascading Removal

5.1 Introduction

Information/influence propagation in social networks is a phenomenon where innovations, opinions, ideas, recommendations or tendencies become widely accepted and adopted due to the influence or importance of a few nodes (Bikhchandani, Hirshleifer, & Welch, 1992; Goldenberg, Libai, & Muller, 2001; Granovetter, 1978). For instance, opinions or rumors either disappear quickly or propagate and create considerable impact on the targeted networks. Node disappearance may also spread from one node to another across the network creating cascading behavior. The cascading phenomenon is mainly analyzed either by identifying the most influential nodes according to their features in the network, detecting quickly the phenomenon or targeting a minimum set of nodes that could maximize the spread of influence or minimize the propagation of a rumor or an outbreak in the network.

In this chapter we propose a procedure which is inspired from propagation models (Kempe et al., 2003) to predict the nodes that would disappear once one or many nodes vanish through a cascading effect. Prediction is modeled based on an extensive set of network features including node, edge and community features. The cascading removal phenomenon is imitated in the present work by three well-known influence maximization cascading models in addition to two variants of our cascading strategy which sounds more consistent with human behavior over cascading removals. In our strategy, we propose a new formula in which the weight of a node is not assigned solely based on node degree but according to a set of parameters that take into account the orientation and weight of links (see Subsection 5.2.2). Such a formula can be adapted to a customized/optimized equation that better fits the nature and the underlying semantics of the social network under study. The prediction is done for an individual iteration of the cascading models, with the ability to be generalized on the entire course of cascades without any loss of generality.

To validate our approach we compare the prediction accuracy over three real-life networks and five synthetically generated structures that imitate real social networks.

5.2 Proposed Approach

Our approach can be considered related to the general framework of outbreak detection since we attempt to identify the set of nodes in a network that are affected by the

deletion of one or many nodes that we call *triggers* in a similar way a virus or information is spread over the network and reaches such affected vertices.

In social network analysis, there are several key measures (CORE, 2012; Ilyas & Radha, 2011; Krebs, 2011; Wasserman & Faust, 1994) to determine nodes with important positions and possibly maximal spread influence. Good examples of those are degree, betweenness and closeness centralities as presented in Chapter 2. We have already used centrality measures to capture and exploit the position of nodes to reorganize a network once a node disappears (see Chapter 4). We believe that there are potentially other features that significantly contribute to the removal cascading behavior of nodes. Finding those features is an obvious need to help us model the cascading removal in a social network.

As indicated earlier, centrality measures express the importance of nodes in a network. The weight of a node or an edge may also be regarded as its degree of influence. Based on the strength of the influence, a node can propagate its effect among the neighbors through the edges. Therefore, the number of neighbors as well as the size and the number of communities are also other potential features to trigger cascading behavior. Node i can potentially influence its neighbors. Likewise, the fast spread of influence depends on the number and size of communities a node belongs to. That is the reason we generate synthetic networks using parameters related to the presence and size of communities as seen later in Subsection 5.4.1.

Since conducting any research on the cascade removal problem needs to have node actions over time (with time stamps) rather than a network snapshot, it is not an easy task to obtain the required data. Hence, the validation of our approach is conducted using five cascading algorithms (including the two variants of our procedure) that imitate the failure cascading effect in networks as well as five synthetically generated networks, two real networks found in Newman`s site (M. E. J. Newman, 2013) and one real network data from Openflights website (Openflights). Thereafter, the goal of our work is to find a procedure to predict the cascading behavior based on a set of features.

5.2.1 Network features

As already mentioned, the main goal of this research is to show to what extent the cascading removal effect in social networks can be analyzed and/or modeled at individual iterations. In other words, without losing the generality, if we find a pattern that in some degree can represent the cascading deletion effects (at the first step of the cascade), it can be projected to the other iterations likewise. In the sequel we simplify the entire problem of cascading node removal detection/modeling to identify nodes to be affected by the initial deleted nodes at the first level of the cascade phenomenon. In the rest of the chapter we

present an approach to model and predict an individual step of cascading deletions once one or many nodes disappear from the network.

To perform the above task we need to have a wide quantitative analysis of the network as a primitive requirement of any pattern recognition procedure. We therefore conduct the quantitative social network analysis (focused on cascade deletion of nodes) through a variety of network features before and after initial node deletions. Among those features there are indegree/outdegree, betweenness centrality, closeness centrality, PageRank, transitivity, authority, eccentricity, and similarity of nodes. An extensive list of one hundred engineered features for each generated social network has been extracted to be used for the cascade modeling/prediction process as shown in the appendix.

5.2.2 Cascading strategies

Our cascade prediction approach is designed and implemented based on a variety of five different removal cascading strategies with the following questions in mind:

- To what extent, cascading phenomenon in the different chosen strategies can be captured by learning models?
- What are the factors that have an impact on cascading removal predictability in a social network?

The selected models are: (i) Linear Threshold Cascade, (ii) Independent Cascade, (iii) Weighted Cascade, (iv) Random-intuitive Cascade and (v) Intuitive-threshold Cascade models. Since the first three cascading models (see Kempe *et.al* (Kempe et al., 2003)) are already defined and referred in Subsection 3.2 of Chapter 3, we only describe in this chapter the "random-intuitive" and "threshold-intuitive" model variants that we propose.

We add the above two cascade strategy variants in a more intuitive and comprehensible manner in order to observe the predictability measure when the cascading removals happen. To that end, we propose the following weighting equation to be applied to nodes in a directed and weighted network. Such formula can be adapted to a customized/optimized equation that better fits the nature and the underlying semantics of the social network under study.

The Random-Intuitive Cascade strategy we define is a kind of *"Weighted Cascade Model"* in which the node weights are not assigned solely based on node degree but according to the following equation:

$$\text{Equation (1)} : W_i = \frac{a \times b \times c}{x \times y \times z}$$

where w_i is the weight of cascade removal effect for node i of the network and hence represents the probability to activate node i after the disappearance of other nodes in the

network. It is important to note that cascading removal affects *solely* nodes that have at least *an incoming link from a deleted node*. The parameters have the following meaning. a is the number of adjacent deleted nodes, b is the average weight of outgoing links of all adjacent deleted nodes to node i, c is the average weight of incoming links before adjacent deletions, x is the indegree of node i before adjacent deletions, y is the outdegree of node i after adjacent deletions and z is the average weight of incoming links after adjacent deletions. The parameters, a, b, c, x, y and z are normalized. The value of x will never be equal to zero because w_i is computed only for nodes that have an incoming link from the deleted node. The value of y and z is changed to a small value one ε (*e.g.* 0.1) whenever it is equal to zero to prevent the value ∞ for w_i. Intuitively, a zero value for y or z means that the corresponding node has a high probability to be affected by the cascading removal because it is linked to deleted nodes only. The cascading removal has an effect on node i if the normalized value of wi is at least equal to the random value r where $0 < r \leq 1$.

It is important to note that the values of a, b and c are selected to be in the nominator since they have a direct impact on the weight of a node while the values of x, y and z are selected to be in the denominator because they have an inverse relation with the weight value.

We propose the Intuitive-Threshold Cascade strategy in a similar way as the random-intuitive cascade strategy except that the condition of the cascading effect assumes that the weight value w_i is greater than or equal to a predefined threshold t instead of a random value r. In our experiments, the value 0.5 has been selected as the threshold.

5.2.3 An illustrative example

To illustrate the step of identifying activated nodes, let us consider a simple connected, directed and weighted network of fourteen nodes as shown in Figure 41 and Table 13.

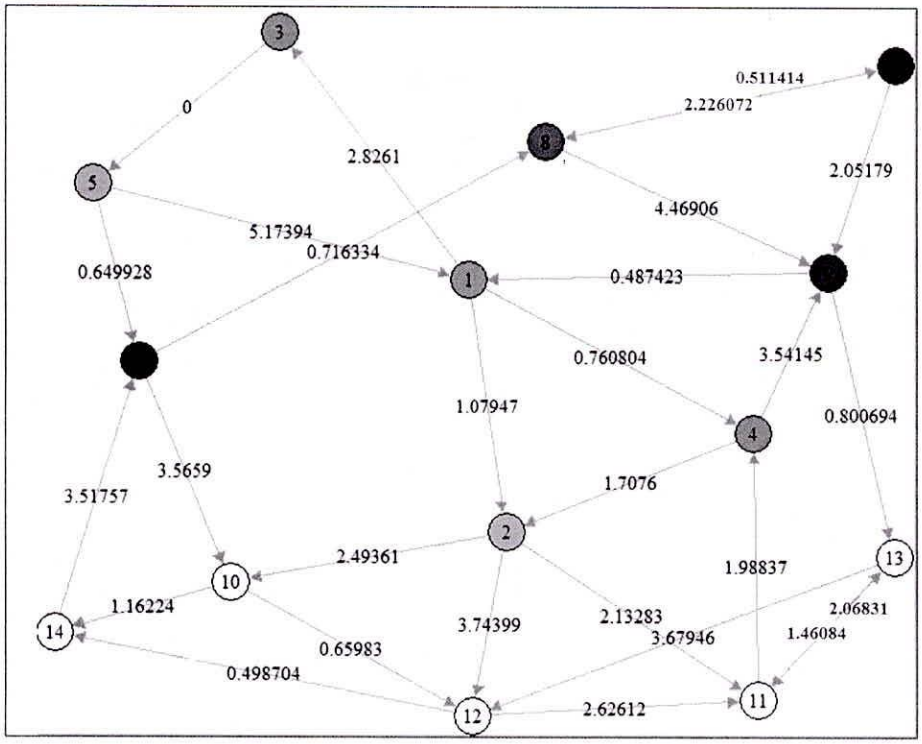


Figure 41. Network example

Let us assume that nodes 2 and 13 are deleted from the network. Our objective is then to know the impact of such deletions on the remaining nodes and more specifically to predict the nodes that will disappear.

From Figure 41, one can see that only nodes 10, 11 and 12 may be affected by the deletion of nodes 2 and 13 because they have incoming links from at least one of the deleted nodes.

Table 13. The weights of network example

Source node	Target node	Edge weight
1	2	1.07947
1	3	2.8261
1	4	0.760804
5	1	5.17394
5	6	0.649928
7	8	2.26072
7	9	2.05179
2	10	2.49361
2	11	2.13283
2	12	3.74399
3	5	0
6	10	3.5659
6	8	0.716334
4	2	1.7076
4	9	3.54145
13	11	2.06831
13	12	3.67946
10	14	1.16224
10	12	0.65983
8	7	0.511414
8	9	4.46906
14	6	3.51757
11	4	1.98837
11	13	1.46084
9	1	0.487423
9	13	0.800694
12	14	0.498704
12	11	2.62612

As an illustration, let us compute the value of w_i for Node 11 by providing details about the calculation of parameters a, b, c, x, y and z. The number a of adjacent deleted nodes is equal to 2. The average weight b of all outgoing links from nodes 2 and 13 to node 11 is (2.13283+2.06831)/2 = 2.10. The average weight c of outgoing links of all adjacent deleted nodes is equal to (2.13283+2.62612+2.06831)/3 = 2.28. The indegree x of node 11 before deletion is 3 whereas its outdegree y after adjacent deletion is 1. The average weight of incoming links to Node 11 after deletion is 2.63. Therefore, the assigned weight to node 11 is 1.22 while its normalized value with respect to the (minimal and maximal) value of other

nodes in Table 13 is 0.15. Using a threshold of 0.5 for example, only node 12 is a candidate for deletion since its normalized value of w_i is 1.

Table 14 provides the value of parameters in Equation 1 as well as the absolute and normalized values of w_i for the identified nodes.

Table 14.The computed weight of the candidate nodes for removal

Node i	Number of adjacent deleted nodes	Average weight of outgoing links from adjacent deleted nodes to node *i*	Average weight of incoming links to node *i* before deletion	Indegree of node *i* before adjacent deletions	Outdegree of node *i* after deletion	Average weight of incoming links after adjacent deletion	Non normalized w_i	Normalized w_i
	a	***b***	***c***	***x***	***y***	***z***		
10	1	2.49	3.03	2	2	3.57	0.53	0
11	2	2.10	2.28	3	1	2.63	1.22	0.15
12	2	3.71	2.69	3	2	0.66	5.04	**1**

5.3 Procedure Validation

In order to validate the two variants of our approach and compare their accuracy against three existing models, we propose the following assessment procedure which starts from a given social network $G = (V, E)$ to return a set of nodes that are most predictable to be influenced by a randomly generated set of deleted nodes. The prediction accuracy of the compared models is also provided.

Step 1 – Select initial trigger nodes

- Compute a set of most discriminating network features for each node $n \in V$ (see the first 22 features in the appendix). Such features cover the main attributes of nodes before vertex deletion and are the most discriminating ones based on the application of the Chi-squared test statistic.

- Run clustering sampling on the nodes with their features to randomly select trigger nodes, *i.e.*, the nodes that trigger the deletion of other vertices. Expectation Maximization (EM) algorithm is executed with a maximum of 100 iterations and 1.0E-6 as minimum standard deviation. The number of clusters is then determined through the cross validation procedure. A random subset T of 4 to 8% of nodes from each cluster is selected to represent trigger nodes.

Step 2 – Identify the nodes to be deleted

- Run the five cascading strategies described earlier to identify the sets A_i ($1 \leq i \leq 5$) that represent the subsets of nodes affected by the trigger nodes $t \in T$. These subsets A_i are predicted via learning algorithms.
- Calculate the set of 100 features (see the appendix) for every node $d \in V$ that has an incoming link from any deleted trigger node $t \in T$.

Step 3 – Compute the accuracy of the five models

- We took Bayesian logistic regression as a statistical method for classification while we used SVM as a mathematical one. We train and test these two classification models, using the 100 features for every node d ∈V in order to classify every node ai ∈ Ai for the five cascading strategies.

- Divide the total number of correctly classified instances in the test set by the total number of test set instances in order to evaluate the prediction accuracy of the trained models.

The normalized value of a given feature is computed as follows: $(w_i - Min_w)/(Max_w - Min_w)$, where w_i is the value of a given feature while Min_w and Max_w represent their minimal and maximal values respectively.

One can see that the complexity of the above procedure is in linear order of $n = |V|$. However, the procedure is also influenced by all the sub-processes applied in the algorithm *e.g.*, EM which is in $O(n*k*i)$ where k is the number of clusters and i is the number of iterations of the EM algorithm. Computational complexity of the classifiers is also a matter that is directly related to the classifier to be chosen and the implementation version of the algorithm.

Moreover, computational complexity of each one of the features used in this algorithm is also another influential factor that was discussed in Chapter 2. Table 14 in Subsection 5.4.1 gives an idea about the execution time (in seconds) needed to compute 100 features for the three real networks. However, we believe that a small subset of those features is enough to correctly describe nodes and their importance inside the network.

5.3.1 Cascading triggers

After selecting the cascading strategies we need to have a set of trigger nodes to start the cascading removals. The set of initial deletions is supposed to be a small subset of the entire

network. It includes nodes with different feature values. Although at a first glance the subset can be selected via random sampling, there would be a higher probability for typical nodes (nodes with features values in average range) to be selected rather than some specific nodes with lower frequencies. However, the portion of very influential nodes (*e.g.*, leaders and mediators) in a social network is far lower than that of typical nodes. The former group with their degree of influence is more interesting for different research purposes compared to the latter one. This intuition leads us to select cluster sampling (J. E. Cohen, 1997) instead of random sampling to get a representative sample of the initial set of deleted nodes as starters for each one of the selected cascading models.

In the cluster sampling technique, the set of nodes is divided into groups (or clusters) with similar features. Then, the required samples are collected by random sampling of the nodes within each group (or cluster). We performed Expectation Maximization (EM) algorithm (Dempster, Laird, & Rubin, 1977) with maximum 100 iterations and 1.0E-6 as minimum standard deviation to explore the node clusters of the networks. The number of clusters is determined through the cross validation procedure. A subset of nodes in each cluster was selected as cascading removal triggers through the cluster sampling technique. Those subsets are then used in the five listed cascading strategies to calculate the first level of cascading deletions accordingly. Such deletions are the target (class labels) of our learning algorithm in the next step.

At the final stage of our validation procedure we run a variety of pattern recognition algorithms on the training part of the extracted and engineered features integrated with the identification of the cascade triggers and their first level targeted cascading removals. The explored patterns are then evaluated based on their power of cascading removal prediction over the testing part of the above data.

5.3.2 Removal prediction accuracy

At the last step of our cascading effect analysis we apply the explained features in order to train two well-known classifiers. The first algorithm is Bayesian Logistic Regression (Genkin, Lewis, & Madigan, 2012) which is regarded as a statistical modeling approach whereas the second method is the kernel based Support Vector Machine (SVM) algorithm (Platt, 1999).
By relying on Weka machine learning system (Hall et al., 2009), we used SMO implementation based on the Polynomial Kernel for the SVM classifier while we applied the latest stable version (3.6) of the Bayesian Logistic Regression with its default parameter values. The classification task is to label neighbors of the cascading trigger nodes as either "remains" or "vanishes" after the deletion of the trigger nodes.

The performance of the trained classification models is evaluated over tenfold cross-validation processes. This means that the classifier is trained on nine parts of the data and tested on the remaining part. Then, the process is repeated ten times for different splits, and the results are averaged over the ten folds. The baseline accuracy is given by a classifier that always guesses the majority class. In our data set, normally the number of activated nodes (*i.e.*, the nodes that are deleted through the cascading process) is lower than the number of non-activated nodes. In other words, the number of deleted neighbors targeted by the cascading effect is virtually less than 50% of the number of neighbors of the trigger nodes. Since the unbalance distribution can influence the classification performance (*i.e.*, creates a bias on the more frequent class of non-activated neighbors) and its baseline, we decided to apply the under-sampling technique to have a balance binary classification task over a variety of models and classifiers. This strategy unifies the classification baseline to a value equal to 50% for all experiments, therefore making possible and more reliable the comparisons and consequently inferences over networks with different sizes and a variety of cascading strategies.

5.4 Empirical Study

5.4.1 Data set

To conduct any analysis or modeling on cascading behavior we need to have some snapshots of networks at different time points. For our purpose, any two ordered snapshots are useful when they are extracted closely, before and after initial acts of cascade activators, *i.e.*, initial deleted nodes. We understand that initial deletions gradually happen on a network. However, without loss of generality we assume that in any interval between two consecutive snapshots of a network there is a set of initial deletions that occur. Those snapshots can give us valuable information about changes in structure and features of the network upon the deletions that potentially can be recognized or modeled as patterns of the cascading effects.

The empirical study concerns two types of data: real networks and synthetically generated ones. In the former case we used three networks, namely *Netscience* (Newman, 2001) and *Hep-th* network (M. E. J. Newman, 2013) as well as *Route database* (Openflights). *Netscience* network concerns co-authorship of scientists who were working on network theory and experiment. This weighted network contains 1,589 scientists and 2,742 collaborations. *Hep-th* network covers co-authorship between scientists that have preprints on the High-Energy Theory E-Print Archive between Jan 1, 1995 and December 31, 1999. It is also a weighted network which contains 8,361 scientists and 15,751 collaborations. In these two networks, an edge between two nodes (authors) expresses a joint publication.

Route database describes 59,036 airline routes (edges) between more than 3,200 airports (nodes). To get a directed and weighted network, we first identify the existing airports and then add a directed edge from a source to a destination with a relative weight that corresponds to the frequency of the route taken by the airline companies. The obtained network contains 3,282 airports and 34,354 directed edges (connections).

Table 15. Number of trigger nodes and their neighbors for the three networks

	Netscience		**Hep-th**		**Route**	
	|V|=1589	**|E|=2742**	**|V|=8361**	**|E|=15751**	**|V|=3282**	**|E|=34354**
Number of trigger nodes	136		303		270	
Number of neighbors of trigger nodes that have incoming links	145		462		1,238	

The execution time needed for computing the algorithm before and after node deletion can be seen in Table 16 below.

Table 16. Execution time for real-life social networks (in milliseconds)

	Netscience	**Hep-th**	**Route**
Feature computation before trigger deletion	0.23	0.66	0.26
Feature computation after trigger deletion	0.67	4.0	4.72

Besides giving the size of each network, Table 15 exhibits the number of selected trigger nodes (see Step 1 of the validation procedure), the number of nodes that have incoming links from trigger nodes as well as the execution time needed to compute the 22 features for the whole set of nodes and then the 100 features of the nodes targeted for deletion. *Hep-th* network has the highest execution time because it has the biggest number of nodes but fewer edges than *Route*. The execution time required to compute features before deleting trigger nodes is slightly lower compared to the one after deletion because in the former case we use only 22 features on the whole network while in the latter case we compute the 100 features for the nodes that have incoming links from trigger nodes.

In order to have observations on a variety of network sizes we also decided to run our experiments on simulated synthetically generated networks. To produce those networks

similar to real social networks we relied on related literature around characteristics of social networks (Ferrara, 2012; Gupta, Garg, Mahanti, Carlsson, & Arlitt, 2009; Nguyen, Dinh, Nguyen, & Thai, 2011; Schneider et al., 2013; Wicklin, 2011). According to the above studies and regarding the size of generated networks we tuned the Lancichinetti's network generator parameters (Lancichinetti & Fortunato, 2009) to generate ten directed and weighted social networks with the following values[3]:

- *N* (number of nodes) =1000 to 10000 and more precisely 1000, 2000, 3000, ..., 10000
- *k* (average node degree) = 10
- *maxk* (maximum degree) = 100
- *mut* (mixing parameter for the topology) = 0.23. This parameter represents the average proportion of links of a node that are outside its community.
- *muw* (mixing parameter for the weights) = 0.23
- *beta* (exponent for the weight distribution) =1.5
- *t1* (minus exponent for the degree sequence) = 2
- *t2* (minus exponent for the community size distribution) = 1
- *minc* (minimum for the community sizes) =10
- *maxc* (maximum for the community sizes) =30
- *on* (number of overlapping nodes) = 0.3*N
- *om* (number of memberships of the overlapping nodes) =3.

We selected five out of the ten initially generated networks. The selected network sizes are 2000, 4000, 6000, 8000 and 10000 nodes.

5.4.2 Environment

The experiments were conducted on an Intel Core i5 with 4 GB of RAM and 3.10 GHz running under Windows 7 Professional. After computing one hundred features using R language, we run the first step of the validation process to get trigger nodes by using the Weka machine learning system (Hall et al., 2009), and mainly the feature selection method based on Chi-squared test statistic.

The second step about the five cascading strategies is implemented in Java while the third one exploits the functionalities of Weka to estimate the classification accuracy of the two classifiers. More precisely, we used SMO implementation using

[3] Parameters in the appendix are listed to help the reproducibility of this research work.

Polynomial Kernel for the SVM classifier while we applied the latest stable version (3.6) of the Bayesian Logistic Regression with its default parameter values.

5.4.3 Experimental results and discussion

In this part we compare the prediction accuracy of different cascading strategies using two classification algorithms by varying the network size. As already mentioned a subset of about 7% of the total number of nodes for each network was selected as cascading removal triggers through the cluster sampling technique.

Table 17. The accuracy of cascading removal prediction for synthetically generated networks using SVM classifier

Cascading Strategies	Network Size				
	2,000 nodes	4,000 nodes	6,000 nodes	8,000 nodes	10,000 nodes
Linear Threshold	70.1031	69.6875	72.7941	70.3349	68.677
Independent Cascade	72.0679	72.8171	71.7472	72.6309	71.6684
Weighted Cascade	73.2143	69.4444	70.7031	71.9008	73.2394
Random-Intuitive	75.3086	72.6744	76.7544	76.7123	73.6842
Intuitive-threshold	**79.8859**	**81.0562**	**82.5041**	**84.1212**	**85.859**

Table 18. The accuracy of cascading removal prediction for synthetically generated networks using Bayesian Logistic Regression classifier

Cascading Strategies	Network Size				
	2,000 nodes	4,000 nodes	6,000 nodes	8,000 nodes	10,000 nodes
Linear Threshold	72.1649	71.25	71.0784	70.8134	70.0389
Independent Cascade	70.2247	70.2636	71.0657	72.8803	70.0407
Weighted Cascade	74.1071	69.4444	71.4844	70.2479	73.5915
Random-Intuitive	74.6914	73.8372	74.5614	74.6575	74.5614
Intuitive-threshold	**80.2081**	**81.5843**	**83.7236**	**84.2424**	**85.3333**

Table 19. The accuracy of cascading removal prediction for real networks via SVM classifier

Cascading Strategies	Networks		
	Netscience	Hep-th	Route
Linear Threshold	68.1732	69.6613	69.3194
Independent Cascade	70.3341	70.7869	71.6821
Weighted Cascade	69.5134	69.9348	70.4380
Random-intuitive	72.7216	73.5951	73.3151
Intuitive-threshold	**76.1039**	**79.1001**	**80.8832**

Table 20. The accuracy for real networks via Bayesian Logistic Regression classifier

Cascading Strategies	Networks		
	Netscience	Hep-th	Route
Linear Threshold	70.1948	69. 3947	70.8521
Independent Cascade	70.7402	70.3958	72.0867
Weighted Cascade	69.1071	68.8094	70.4641
Random-intuitive	72.01799	72.8094	74.2121
Intuitive-threshold	**76.8812**	**78.9134**	**81.2113**

Overall, our approach gives the best accuracy for both synthetic and real networks using the two classifiers. For example, the results in Table 17 related to synthetic data analysis using SVM classifier show that the accuracy is between 68.67 and 76.88% for the first three cascading strategies in which there is a random function on cascading removal propagation while the intuitive-threshold strategy variant gives an accuracy between 79.88 and 85.85%.

We note that including intuitive removal propagation factors in the cascading model along with the role of random function can slightly increase the accuracy (almost 76%) in "Random-intuitive" strategy. However, the random function innately limits the improvement of any learning method predictability. With intuitive-threshold, the accuracy is the best one and increases with the number of nodes (see Tables 17 and 18) and reaches almost 86%.

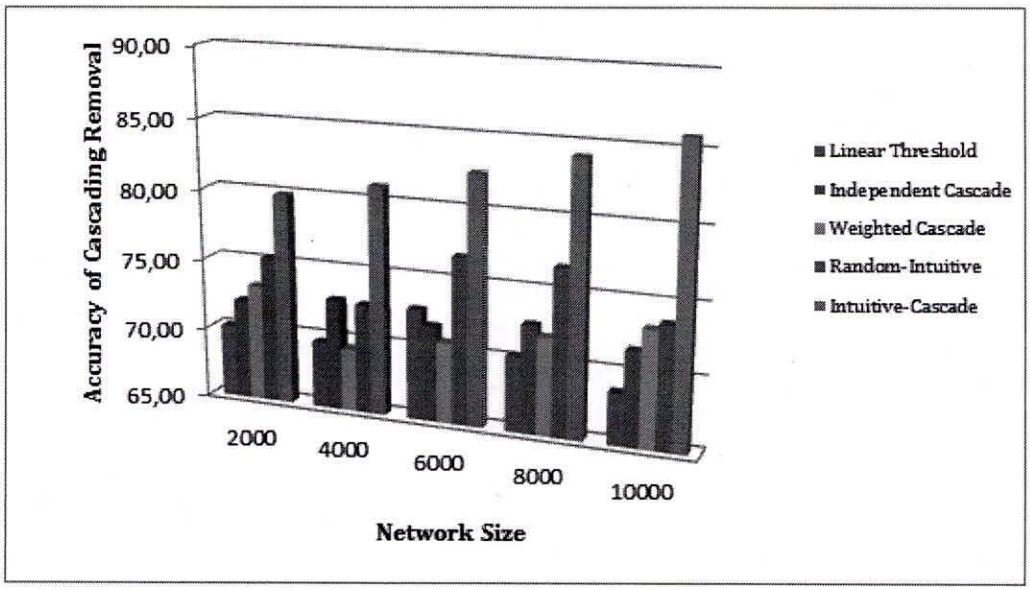


Figure 42. The accuracy of cascading removal prediction of synthetically generated networks via SVM classifier

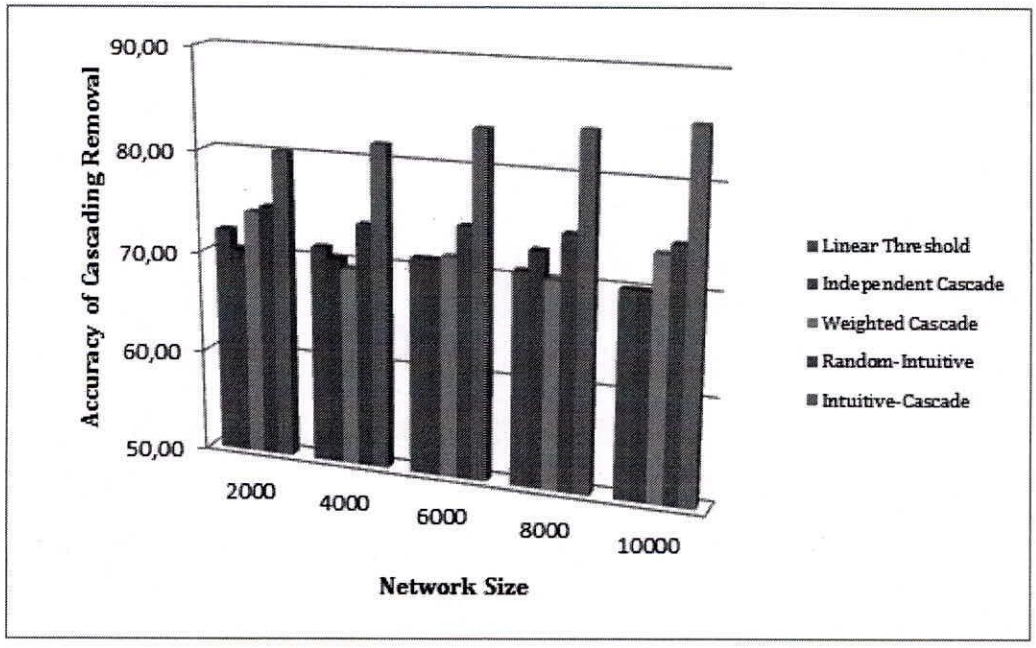


Figure 43. The accuracy of cascading removal prediction of synthetically generated networks via Bayesian Logistic Regression classifier

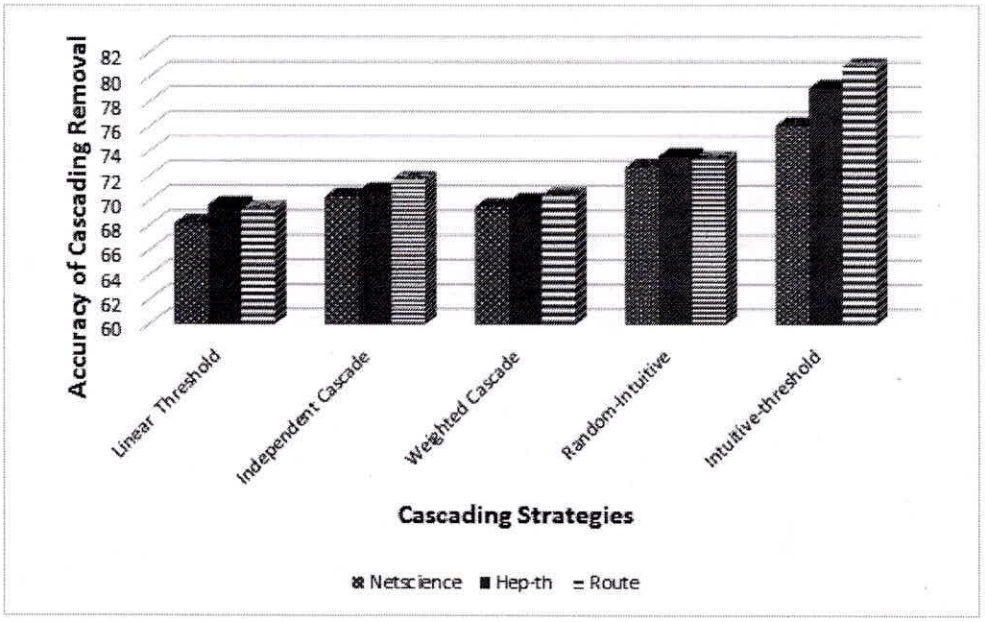


Figure 44. The accuracy of cascading removal prediction of real networks via SVM Classifier

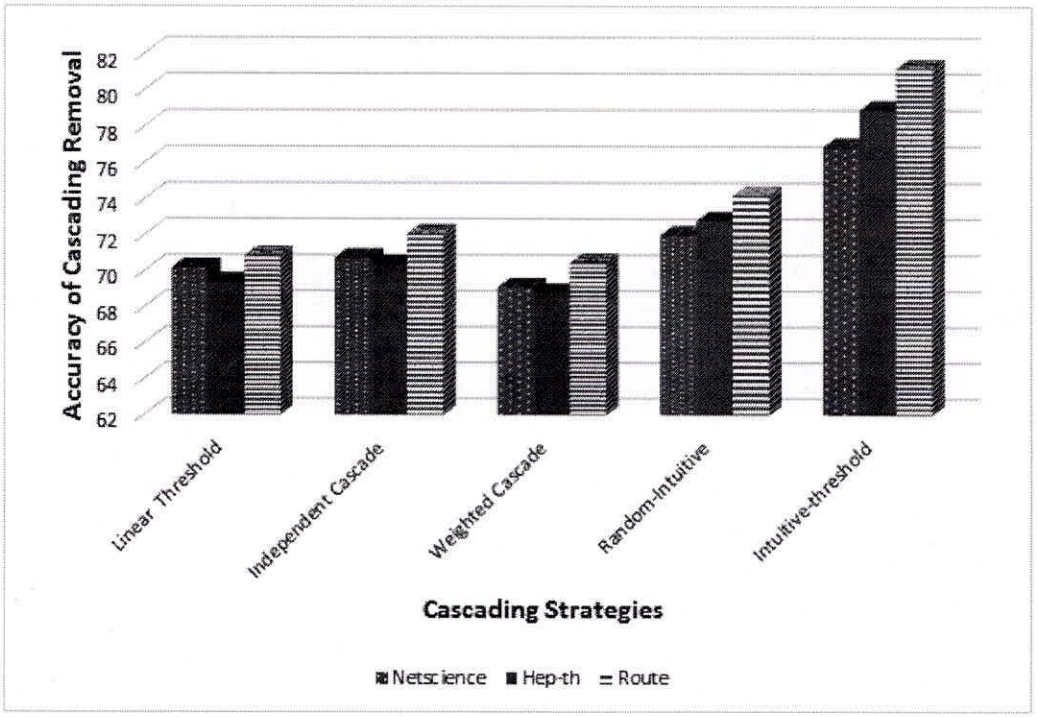


Figure 45. The accuracy of cascading removal prediction of real networks via Bayesian Logistic Regression Classifier

Indeed, the results for the *"Intuitive-threshold"* strategy in which node removal effects are propagated based on intuitive factors show that (i) the accuracy of prediction can be improved when the number of nodes increases[4] and (ii) the prediction can be done more accurately (almost 86%) where the removal propagation phenomenon is effective upon recognizable intuitions. The accuracy improvement is also observed for real networks as depicted by Tables 19 and 20 and Figures 44 and 45.

Considering the experimental results and the rationale behind the proposed approach, we believe that our solution mostly applies when the randomness behavior is not a dominating attribute among the nodes, which is mostly the case in real social networks. It will also work when the average degree of nodes and/or the a priori threshold are not so high to allow the system to predict a small number of affected nodes among tens or even hundreds candidate neighbors. However, if the limit of the number of affected nodes is relaxed, the performance of the suggested approach would not be affected. Regarding the computational costs of the method we recall that the major load is on the feature calculation part which can be done gradually (incrementally) through the growth of the network and be updated periodically according to the volume of the changes.

As another option to moderate the aforementioned cost we can reduce the number of features using a feature selection procedure over a sample data and then build the prediction models based on the top 20 or 30 selected features rather than one hundred features. It is notable that the performance of the prediction process can be improved by tuning and customizing the best features and also classifier parameters in the presence of a specific network. For instance, the a priori threshold used to select triggering nodes according to their weight can be searched through an optimization algorithm.

5.5 Conclusion

In this chapter we have presented an approach to study the predictability degree of the cascading effect/failure focused on the cascading effect of node removals. We trained two well-known machine learning algorithms, namely SVM and Bayesian Logistic Regression based on a set of one hundred network features, including node and path features. We compared the prediction accuracy over five synthetically generated networks as well as three real social networks. The cascading removal phenomenon was simulated by three influence maximization cascading models in addition to two variants of our cascading strategy, which are designed based on human behavior. We achieved the prediction accuracy for *the intuitive-*

[4] This attribute can be also justified by increasing the number of training data.

threshold strategy in the range [79.88% - 85.85%] for synthetic data and [76.1% - 81.2%] for the three real networks after one iteration of the cascading models. The accuracy of the three cascading models are within the interval [68.67% - 76.75%] for synthetic networks and [68.17% - 72%] for real ones. This indicates that our approach and mainly the intuitive-threshold variant is more accurate than the three known cascading methods.

This work has potential applications in different domains such as viral marketing and customer/ employee attrition. In the first case, it will help predict the propagation of information about a new product or service by network actors to the rest of the structure while in the latter case, it could help forecast the churn of customers following the departure of a few of them.

6. Conclusion and Future Work

The goal of this thesis is to analyze the impact of node disappearance on a social network structure and aims mainly at studying two topics in social network analysis: (i) network reorganization following a node disappearance, and (ii) cascading removal following the deletion of one or many nodes from the network. For the first issue, we have proposed an approach that seeks for a substitute node whenever the disappearing node plays an important role in the network either as a leader, mediator or witness with a high score of degree, betweenness, and closeness centrality, respectively. Links between the substitute and other nodes in the network are then added according to the needs of the user or the application in order to maintain the stability of the network and avoid disconnectivity. For the second issue, we have proposed two variants of a cascading effect method to identify the nodes that could disappear from a weighted and oriented network once one or many nodes are deleted.

The first topic exploits three centrality measures to predict the future structure of the related graph after the disappearance of one node. It uses the network topology to possibly find the substitute to the deleted node whenever the latter is central (leader, mediator or witness) and add new links between identified nodes in order to maintain the network stable and connected. To analyze the impact of the node deletion and the subsequent network restructuring we capture the overall picture of the network before and after the update using known network measures such as clustering coefficient, network diameter efficiency and so on. An empirical comparison of our approach against two related ones is also provided and shows its strengths and weaknesses with respect to network stability, execution time and accuracy.

The second topic concerns the prediction of a deletion in cascade (snowball/cascade effect) when a set of nodes disappear from a directed and weighted network. From real-life situations, the disappearance of nodes or links can propagate and create a considerable impact on the network. The study of such cascading phenomenon can hence be helpful to an organization to predict the attrition of a group of its employees or customers when a central individual leaves the organization or to a social network manager to let him identify the individuals who may leave the network following the departure of a key actor.

Our approach proposes two variants of cascading models in which a non-trivial weight is assigned to the neighbors of deleted nodes with incoming links from the disappearing nodes. The candidate nodes for deletion are those with a weight greater than or equal to a given threshold. To validate our approach, we trained two well-known machine

learning algorithms, namely SVM and Bayesian Logistic Regression based on a set of one hundred network features, including node and path features. We compared the prediction accuracy over five synthetically generated networks as well as three real social networks. The cascading removal phenomenon was simulated by three influence maximization cascading models in addition to the two variants of our cascading strategy.

Our future work covers the following topics:

- Extend and adapt our work on the two topics to link rather than node disappearance. For example, in the *Route network* used in the empirical study, it is more realistic to observe the deletion of one or many connections between two given airports than the deletion of an airport and afterwards forecast the impact of route removal on other routes used by airline companies and flights.
- Analyze the impact of exploiting the presence of communities on network reorganization following a node disappearance and the search for a substitute.
- Define a new approach towards criminal network disintegration by first identifying communities and key players to further remove the important nodes and predict the node deletions that will likely follow by applying and adapting our approach for cascading removal.

Appendix

List of 100 network features that have been engineered, calculated and used in this paper:

1. Average incoming weights before deletion.
2. Minimum incoming weights before deletion.
3. Maximum incoming weights before deletion.
4. Average outgoing weights before deletion.
5. Minimum outgoing weights before deletion.
6. Maximum outgoing weights before deletion.
7. PageRank of the node before deletion.
8. Transitivity of the node before deletion.
9. Indegree of the node before deletion.
10. Outdegree of the node before deletion.
11. Degree centrality before deletion.
12. Betweenness centrality of the node before deletion.
13. Closeness centrality of the node before deletion.
14. Eigenvalue of the node before deletion.
15. Coreness[5] value of the node before deletion.
16. Hub value of the node before deletion.
17. Authority value of the node before deletion.
18. Eccentricity value of the node before deletion.
19. Number of community membership of a node before deletion.
20. Average size of community membership of a node before deletion.
21. Minimum size of community membership of a node before deletion.
22. Maximum size of community membership of a node before deletion.

    ----------[6]
23. If the node is a deleted one.
24. Number of neighbors of deleted nodes.

---

[5] Coreness of a node is k if the vertex belongs to the k-core but not to the (k+1)-core We recall that a k-core is the largest subgraph that contains nodes of degree at least equal to k.

[6] The first 22 features that we choose for the first step of the validation process cover the main attributes of nodes before vertex deletion

25. Average incoming weights of all neighbors of deleted nodes.
26. Minimum incoming weights of all neighbors of deleted nodes.
27. Maximum incoming weights of all neighbors of deleted nodes.
28. Average outgoing weights of all neighbors of deleted nodes.
29. Minimum outgoing weights of all neighbors of deleted nodes.
30. Maximum outgoing weights of all neighbors of deleted nodes.
31. Average PageRank of neighbors of deleted nodes.
32. Minimum PageRank of neighbors of deleted nodes.
33. Maximum PageRank of neighbors of deleted nodes.
34. Average transitivity of neighbors of deleted nodes.
35. Minimum transitivity of neighbors of deleted nodes.
36. Maximum transitivity of neighbors of deleted nodes.
37. Average degree of neighbors of deleted nodes.
38. Minimum degree of neighbors of deleted nodes.
39. Maximum degree of neighbors of deleted nodes.
40. Average Degree centrality of the neighbors of deleted nodes.
41. Minimum Degree centrality of the neighbors of deleted nodes.
42. Maximum Degree centrality of the neighbors of deleted nodes.
43. Average betweenness centrality of the neighbors of deleted nodes.
44. Minimum betweenness centrality of the neighbors of deleted nodes.
45. Maximum betweenness centrality of the neighbors of deleted nodes.
46. Average closeness centrality of the neighbors of deleted nodes.
47. Minimum closeness centrality of the neighbors of deleted nodes.
48. Maximum closeness centrality of the neighbors of deleted nodes.
49. Average eigenvalues of the neighbors of deleted nodes.
50. Minimum eigenvalues of the neighbors of deleted nodes.
51. Maximum eigenvalues of the neighbors of deleted nodes.
52. Average coreness values of the neighbors of deleted nodes.
53. Minimum coreness values of the neighbors of deleted nodes.
54. Maximum coreness values of the neighbors of deleted nodes.
55. Average hub values of the neighbors of deleted nodes.
56. Minimum hub values of the neighbors of deleted nodes.
57. Maximum hub values of the neighbors of deleted nodes.
58. Average authority values of the neighbors of deleted nodes.
59. Minimum authority values of the neighbors of deleted nodes.
60. Maximum authority values of the neighbors of deleted nodes.

61. Average eccentricity values of the neighbors of deleted nodes.
62. Minimum eccentricity values of the neighbors of deleted nodes.
63. Maximum eccentricity values of the neighbors of deleted nodes.
64. Average number of common neighbors among the neighbors of deleted nodes.
65. Minimum number of common neighbors among the neighbors of deleted nodes.
66. Maximum number of common neighbors among the neighbors of deleted nodes.
67. Average number of common communities with the neighbors of deleted nodes.
68. Minimum number of common communities with the neighbors of deleted nodes.
69. Maximum number of common communities with the neighbors of deleted nodes.
70. Average size of common communities with the neighbors of deleted nodes.
71. Minimum size of common communities with the neighbors of deleted nodes.
72. Maximum size of common communities with the neighbors of deleted nodes.
73. Average incoming similarity with the neighbors of deleted nodes.
74. Minimum incoming similarity with the neighbors of deleted nodes.
75. Maximum incoming similarity with the neighbors of deleted nodes.
76. Average outgoing similarity with the neighbors of deleted nodes.
77. Minimum outgoing similarity with the neighbors of deleted nodes.
78. Maximum outgoing similarity with the neighbors of deleted nodes.
79. Average incoming weights after deletion.
80. Minimum incoming weights after deletion.
81. Maximum incoming weights after deletion.
82. Average outgoing weights after deletion.
83. Minimum outgoing weights after deletion.
84. Maximum outgoing weights after deletion.
85. PageRank of the node after deletion.
86. Transitivity of the node after deletion.
87. Indegree of the node after deletion.
88. Outdegree of the node after deletion.
89. Degree centrality after deletion.
90. Betweenness centrality of the node after deletion.
91. Closeness centrality of the node after deletion.
92. Eigenvalue of the node after deletion.
93. Coreness value of the node after deletion.
94. Hub value of the node after deletion.
95. Authority value of the node after deletion.
96. Eccentricity value of the node after deletion.

97. Number of community membership of a node after deletion.
98. Average size of community membership of a node after deletion.
99. Minimum size of community membership of a node after deletion.
100. Maximum size of community membership of a node after deletion.

References

Adamic, L. A., & Adar, E. (2003). Friends and Neighbors on the Web. *Social Networks, 25*(3), 211 - 230.

AlHasan, M., Chaoji, V., Salem, S., & Zaki, M. (2006). Link Prediction using Supervised Learning. *In Proc. of SDM 06 workshop on Link Analysis, Counterterrorism and Security*.

AlHasan, M., & Zaki, M. J. (2011). A Survey of Link Prediction in Social Networks *Social Network Data Analytics 2011* (pp. 243-275).

Allen, F., & Gale, D. M. (2000). Financial Contagion. *Journal of Political Economy, 108*(1), 1-33.

Anderson, R. M., & May, R. M. (1992). *Infectious Diseases of Humans* Oxford University Press.

B.West, D. (2001). *Introduction to Graph Theory* (2nd ed.): Prentice Hall.

Barabási, A.-L., & Albert, R. (1999). Emergence of Scaling in Random Networks. *Science, 286* (5439), 509-512 doi: 10.1126/science.286.5439.509

Barabási, A.-L., & Bonabeau, E. (2003). Scale-Free Networks. *Scientific American, 288*, 60 - 69.

Ben-Naim, E., & Krapivsky, P. L. (2007). Addition–deletion networks *Journal of Physics A: Mathematical and Theoretical 40*(30), 8607 doi: 10.1088/1751-8113/40/30/001

Bikhchandani, S., Hirshleifer, D., & Welch, I. (1992). A Theory of Fads, Fashion, Custom, and Cultural Change as Informational Cascades. *Journal of Political Economy 100*(5), 992-1026

Blume, L., Easley, D., Kleinberg, J., Kleinberg, R., & Tardos, É. (2011). *Which Networks are Least Susceptible to Cascading Failures?* Paper presented at the FOCS '11 Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science, Washington, DC, USA

Bonacich, P. (1987). Power and Centrality: A Family of Measures. *American Journal of Sociology, 92*(5), 1170 - 1182.

Borgatti, S. P. (2005). Centrality and network flow. *Social Networks, 27*(1), 55 - 71.

Bouguessa, M., Missaoui, R., & Talbi, M. (2014). *A Novel Approach for Detecting Community Structure in Networks*. Paper presented at the The annual IEEE International Conference on Tools with Artificial Intelligence (ICTAI), St Raphael Resort, Limassol, Cyprus.

Brandes, U. (2001). A faster algorithm for betweenness centrality. *The Journal of Mathematical Sociology, 25*(2), 163-177.

Caldarelli, G. (2007). *Communities and Clustering in Some Social Network*. Paper presented at the NetSci International Workshop and Conference Network Science, New York, USA.

Chase, R., & Borgatti, S. (2012). Social Network Analysis Software - Cultural Domain Analysis Software.

Cheliotis, G. (2010). Social Network Analysis (SNA) including a tutorial on concepts and methods. Retrieved November 17, 2013, from http://www.slideshare.net/gcheliotis/social-network-analysis-3273045

Chen, W., Wang, Y., & Yang, S. (2009). *Efficient Influence Maximization in Social Networks.* Paper presented at the KDD'09, Paris, France. 28 June - 1 July.

Choromanski, K., Matuszak, M., & Miekisz, J. (2013). Scale-Free Graph with Preferential Attachment and Evolving Internal Vertex Structure. *Journal of Statistical Physics, 151*(6), 1175–1183. doi: 10.1007/s10955-013-0749-1

Christakis, N. A., & Fowler, J. H. (2010). Social Network Sensors for Early Detection of Contagious Outbreaks *PLOS One*(5). doi: 10.1371/journal.pone.0012948

Cohen, J. E. (1997). Size-estimation framework with applications to transitive closure and reachability. *Computer and System Sciences 55*(3), 441-453.

Cohen, R., Havlin, S., & Ben-Avraham, D. b. (2003). Efficient Immunization Strategies for Computer Networks and Populations. *Phys. Rev. Letters, 91*(24).

Cohen, R., Havlin, S., & Ben-Avraham, D. b. (2006). Structural Properties of Scale-Free Networks. In S. Bornholdt & H. G. Schuster (Eds.), *Handbook of Graphs and Networks: From the Genome to the Internet* (pp. 85 - 109): Wiley.

CORE. (2012). Bridges and Brokers in Social Network Analysis. Retrieved March 15, from Defense Analysis Department at the Naval Postgraduate School http://lhproject.info/core-lab/

Csardi, G., & Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal, Complex Systems 1695*.

Cui, P., Jin, S., Yu, L., Wang, F., Zhu, W., & Yang, S. (2013, 11-14 August). *Cascading Outbreak Prediction in Networks: A Data-Driven Approach.* Paper presented at the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD, Chicago, IL, USA.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological), 39*(1), 1-38.

Domingos, P., & Richardson, M. (2002). *Mining knowledge-sharing sites for viral merketing* Paper presented at the Proceedings of the 8th ACM SIGKDD Conference on Knowledge Discovery and Data Mining.

Easley, D., & Kleinberg, J. (2010). *Networks, Crowds, and Markets: Reasoning About a Highly Connected World* Cambridge University Press.

Everett, M. G., & Borgatti, S. P. (2005). Extending centrality. In P. J. Carrington, J. Scott, & S. Wasserman (Eds.), *Models and Methods in Social Network Analysis* (pp. 57 - 76): Cambridge University Press.

Ferrara, E. (2012). A large-scale community structure analysis in Facebook. *EPJ Data Science, 1*(9).

Fortunato, S. (2010). Community detection in graphs. *Physics Reports, 486*(3 - 5), 75 - 174.

Fortunato, S., Latora, V., & Marchiori, M. (2004). A Method to Find Community Structures Based on Information Centrality. *Physical Review E, 70*(056104). doi: http://dx.doi.org/10.1103/PhysRevE.70.056104

Fowler, J. H., & Christakis, N. A. (2009). Cooperative Behaviour Cascades in Human Social Networks. from http://arxiv.org/abs/0908.3497

Freeman, L. C. (1977). A Set of Measures of Centrality Based on Betweenness. *Sociometry, 40* (1), 35-41.

Friedkin, N. E., & Johnsen, E. C. (1997). Social positions in influence networks. *Social Networks, 19*, 209 - 222.

Genkin, A., Lewis, D. D., & Madigan, D. (2012). Large-Scale Bayesian Logistic Regression for Text Categorization. *Technometrics, 49*(3), 291-304.

Girvan, M., & Newman, M. J. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences, 99*, 7821-7826.

Goldenberg, J., Libai, B., & Muller, E. (2001). Talk of the Network: A Complex Systems Look at the Underlying Process of Word-of-Mouth. *Marketing Letters, 12*(3), 211-223.

Granovetter, M. (1978). Threshold Models of Collective Behavior Author(s). *The American Journal of Sociology, 83*(6), 1420 - 1443.

Gupta, T., Garg, S., Mahanti, A., Carlsson, N., & Arlitt, M. (2009). A Web-based Social Aggregation Service. *ICWSM.*

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explorations, 11*(1).

Hanneman, R. A., & Riddle, M. (2005). *Introduction to social network methods* Retrieved from http://faculty.ucr.edu/~hanneman/nettext/

Hopkins, A. (2010). Graph Theory, Social Networks and Counter Terrorism. Massachusetts: Department of Mathematics. University of Massachusetts Dartmouth. USA.

Huang, X., Vodenska, I., Havlin, S., & Stanley, H. E. (2013). Cascading Failures in Bi-partite Graphs: Model for Systemic Risk Propagation. Massachusetts 02215 USA: Center for Polymer Studies and Department of Physics, Boston University, Boston.

Hussain, D. M. A., & Ahmed, Z. (2010). Dynamical Adaptation in Terrorist Cells/Networks. In K. Elleithy (Ed.), *Advanced Techniques in Computing Sciences and Software Engineering* (pp. 557 - 562). Bridgeport, 06604 U.S.A.: Springer Netherlands.

Ilyas, M. U., & Radha, H. (2011). *Identifying Influential Nodes in Online Social Networks Using Principal Component Centrality.* Paper presented at the Proceedings of IEEE International Conference on Communications, ICC, Kyoto, Japan, 5 - 9 June.

Jamali, M., & Abolhassani, H. (2006). *Different Aspects of Social Network Analysis.* Paper presented at the Proceeding WI '06 Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence Washington, DC, USA.

Kaiser, M. (2008). Mean clustering coefficients: the role of isolated nodes and leafs on clustering measures for small-world networks. *New Journal of Physics, 10*(8), 1 - 11.

Kempe, D., Kleinberg, J., & Tardos, É. (2003). *Maximizing the spread of influence through a social network.* Paper presented at the Proceedings of the 9th ACM SIGKDD Conference on Knowledge Discovery and Data Mining

Kleinberg, J. (2007). Cascading behavior in networks: Algorithmic and economic issues. In Cambridge (Ed.), *Algorithmic Game Theory*. New York.

Knoke, D., & Yang, S. (2008). *Social Network Analysis* (Second ed. Vol. 154): SAGE Publications, Inc

Krause, A., & Guestrin, C. (2009). Optimizing Sensing: From Water to the Web. *Computer, 8*(42), 38 - 45.

Krebs, V. (2011). Social Network Analysis, A Brief Introduction. Retrieved August, 2011, from http://www.orgnet.com/sna.html

Kumpula, J. M., Onnela, J.-P., Samaraki, J., Kaski, K., & Kertész, J. (2007). Emergence of Communities in Weighted Networks. *Physical Review Letters, 99*(22). doi: 10.1103/PhysRevLett.99.228701

Lancichinetti, A., & Fortunato, S. (2009). Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E, 80*(1), 8 pages.

Latora, V., & Marchiori, M. (2001). Efficient behavior of small world network. *Physical Review Letters, 87*(19). doi: 10.1103/PhysRevLett.87.198701

Latora, V., & Marchiori, M. (2003). Economic small-world behavior in weighted networks. *The European Physical Journal B, 32*, 249–263. doi: 10.1140/epjb/e2003-00095-5

Leskovec, J., Kleinberg, J., & Faloutsos, C. (2007). Graph Evolution: Densification and Shrinking Diameters. *ACM Trans Knowledge Discovery Data 1*, 1 - 41.

Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J. M., & Glance, N. S. (2007). *Cost-effective outbreak detection in networks.* Paper presented at the KDD'07 Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining New York, NY, USA.

Leskovec, J., McGlohon, M., Faloutsos, C., Glance, N., & Hurst, M. (2007). *Cascading Behavior in Large Blog Graphs.* Paper presented at the In Proceedings of 7th SIAM International Conference on Data Mining (SDM)

Liben-Nowell, D., & Kleinberg, J. (2003). The Link Prediction Problem for Social Networks. *Journal of the American Society for Information Science and Technology, 58*(7), 913–1067.

Liu, B. (2014). Lecture note on CS 583 - Fall 2014 Data Mining and Text Mining *The PageRank algorithm (of Google).* from http://www.cs.uic.edu/~liub/teach/cs583-fall-14/cs583.html

Manning, C. D., Raghavan, P., & Schütze, H. (2009). *An Introduction to Information Retrieval*: Cambridge University Press, Cambridge, England.

Missaoui, R., Negre, E., Anggraini, D., & Vaillancourt, J. (2013). Network restructuring after a node removal. *International Journal Conference of Web Engineering and Technology (IJWET), 8*(1), 4-26.

Missaoui, R., & Sarr, I. (2015). Social Network Analysis – Community Detection and Evolution, Lecture Notes in Social Networks: Springer.

Negre, E., Missaoui, R., & Vaillancourt, J. (2011). Predicting a Social Network Structure Once a Node is Deleted. *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*, 297 - 304.

Newman, M. E. (2003). Random Graphs as Models of Networks *Handbook of Graphs and Networks, S. Bornholdt and H. G. Schuster (eds.)*. Berlin: Wiley-VCH.

Newman, M. E. (2006). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences, 103*(23), 8577–8582.

Newman, M. E., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E, 69*(2), 026113.

Newman, M. E. J. (2001). Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality. *Phys. Rev. E 64, 016132*. doi: http://dx.doi.org/10.1103/PhysRevE.64.016132

Newman, M. E. J. (2003). The Structure and Function of Complex Networks. *SIAM Review, 45*, 167-256.

Newman, M. E. J. (2004). Fast algorithm for detecting community structure in networks. *Physical Review E, 69*(6).

Newman, M. E. J. (2006). Coauthorship networks and patterns of scientic collaboration. *Physical Review E, 74:036104*.

Newman, M. E. J. (2013). Network data Retrieved 19 April, from http://www-personal.umich.edu/~mejn/netdata/

Nguyen, N. P., Dinh, T. N., Nguyen, D. T., & Thai, M. T. (2011, October 9 -n11). *Overlapping Community Structures and Their Detection on Social Networks.* Paper presented at the IEEE third international conference on social computing (socialcom), Boston, MA.

Okamoto, K., Chen, W., & Li, X.-y. (2008). Ranking of Closeness Centrality for Large-Scale Social Networks. *Frontiers in Algorithmics: Second International Workshop, FAW*, 186 - 195.

Openflights. Airport, airline and route data. http://openflights.org/data.html

Orman, G. K., & Labatut, V. (2009). A Comparison of Community Detection Algorithms on Artificial Networks *Discovery Science* (Vol. 5808, pp. 242-256): Springer.

Platt, J. C. (1999). Fast Training of Support Vector Machines using Sequential Minimal Optimization *Advances in Kernel Methods - Support Vector Learning* (pp. 185 - 208): MIT Press.

Pons, P., & Latapy, M. (2006). Computing communities in large networks using random walks. *Graph Algorithms and Applications, 10*(2), 191 - 218.

Raghavan, U. N., Albert, R., & Kumara, S. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E, 76*(3).

Rochat, Y. (2009). Closeness Centrality Extended To Unconnected Graphs : The Harmonic Centrality Index. *Applications of Social Network Analysis (ASNA)*.

Romero, D. M., & Kleinberg, J. (2010). The directed closure process in hybrid socialinformation networks, with an Analysis of Link Formation on Twitter. *The 4th International AAAI Conference on Weblogs and Social Media*.

Sarr, I., & Missaoui, R. (2013). Managing Node Disappearance Based on Information Flow in Social Networks. *Social Network Analysis and Mining (SNAM), 3*(3), 299-311.

Sarr, I., Missaoui, R., & Lalande, R. (2012). *Dealing with Disappearance of an Actor Set in Social Networks.* Paper presented at the IEEE/ACM International Conference on Advances in Social Network Analisys and Mining, Istanbul, Turkey. 26 - 29 August.

Sarr, I., Missaoui, R., & Lalande, R. (2013). Group Disappearance in Social Networks with Communities. *Social Network Analysis Mining 3*, 651 - 665. doi: 101.1007/s13278-013-0112-1

Schneider, J. A., Cornwell, B., Ostrow, D., Michaels, S., Schumm, P., Laumann, E. O., & Friedman, S. (2013). Network Mixing and Network Influences Most Linked to HIV Infection and Risk Behavior in the HIV Epidemic Among Black Men Who Have Sex With Men. *American Journal of Public Health, 103*(1), e28-e36.

Strogatz, S. H., & Watts, D. J. (1998). Collective dynamics of 'small-world' networks. *Nature, 393*, 440 - 442.

Sun, P. G. (2014). Weighting links based on edge centrality for community detection. *Physica A: Statistical Mechanics and its Applications, 394*, 346 – 357. doi: 10.1016/j.physa.2013.08.048

Toivonen, R., Kovanen, L., Kivel, M., Onnela, J.-P., Saramki, J., & Kaski, K. (2009). A comparative study of social network models: Network evolution models and nodal attribute models. *Social Networks, 31*(4), 240 - 254.

Tsvetovat, M., & Kouznetsov, A. (2011). *Social Network Analysis for Startups: Finding Connections on the Social Web.* Sebastopol, California: O'Reilly.

Tylenda, T., Angelova, R., & Bedathur, S. (2009). *Towards time-aware link prediction in evolving social networks.* Paper presented at the Proceeding SNA-KDD '09 Proceedings of the 3rd Workshop on Social Network Mining and Analysis

Wasserman, S., & Faust, K. (1994). *Social Network Analysis: Methods and Application* (1 ed.): Cambridge University Press.

Wicklin, R. (2011). Modeling Finite Mixtures with the FMM Procedure

Zhuge, H., & Zhang., J. (2010). Topological centrality and its e-Science applications. *JASIST, 61*(9), 1824-1841.

Zio, E., & Sansavini, G. (2011). *Modeling cascading failures in "systems of systems" with uncertain behavior.* Paper presented at the ICASP11-11th International Conference on Applications of Statistics and Probability in Civil Engineering, Zurich, Suisse. August 1-4.