Department of Electrical & Computer Engineering

University of Thessaly, Greece

SPREADING DYNAMICS IN COMPLEX NETWORKS WITH APPLICATIONS IN VEHICULAR NETWORKS

Doctor of Philosophy
in
Electrical & Computer Engineering
By

Pavlos Basaras

November 2017

Dissertation Committee:

Assistant Prof. Katsaros Dimitrios
Professor Manolopoulos Yannis
Professor Tsoukalas Lefteris

The Dissertation of Pavlos Basaras is approved by:

Committee Chairperson

Dept. of Electrical & Computer Engineering,
University of Thessaly, Greece

Dedicated to my family and friends.

Abstract

SPREADING DYNAMICS IN COMPLEX NETWORKS WITH APPLICATIONS IN VEHICULAR NETWORKS

by Pavlos Basaras
Doctor of Philosophy in Electrical & Computer Engineering
Department of Electrical & Computer Engineering, University of Thessaly, Greece
Assistant Prof. Dimitrios Katsaros, Chairperson

With the unprecedented growth during the past decade of different types of social and enterprise networks, alongside naturally occurring networks in human communities, society is on the verge of becoming “fully networked.” Recent advances in information and communications technologies, coupled with the ability to create and store a vast amount of data on various aspects of human behavior, have made it possible to analyze complex networks. Studies range from purely graph-theoretic aspects (size and strength of communities, robustness to attacks, growth models, node connectivity, and so on), to more social-theoretic aspects (for example, homophily and rumor spreading). This research has given rise to computational social science [153] a new field that leverages the ability to collect and analyze data to reveal hidden patterns in individual and group activities. Insights into complex networks’ structural and topological properties have informed work in numerous areas including search engine technology [171] the development of ad hoc network protocols [138] and detecting and containing disease outbreaks [136]. Security researchers have likewise used complex network analysis to study terrorist networks [186] virus propagation over computer networks, and resistance to cyberattacks. Such analyses typically apply graph theory and involve centrality measures, shortest-path algorithms, degree distributions, and so on.

In this thesis we study complex networks from the perspective of network science and graph theory. We employ tools, algorithms and methodologies from the vast literature of network science to deepen our understanding on network topology and network structure, and emphasize on dynamical processes that unfold over complex systems, such as the spreading dynamics. We employ a wide range of popular tools to realize our research interests, and evaluate our proposed techniques, centralities, algorithms, etc., across all the development phases of our work in both real and generated networks. We start by studying the topological characteristics of the network nodes and how topology affects the potential of each node to efficiently spread information in the network. These super spreaders (influential nodes) where traditionally identified by means of their connectivity (degree), i.e., nodes that accumulate more connections are more influential nodes. Additionally the k-shell (or k-core) decomposition of a network exploited several shortcoming of the degree centrality and proved superior in ranking nodes with respect to

their true spreading potential. Nonetheless the k-core requires global knowledge of the network topology and thus is unsuitable for real time applications and dynamically changing networks. Our work introduced a centrality metric, namely *Power Community Index* (PCI), that based solely on local knowledge of a network's connections, outperformed the state-of-the-art competitors in a wealth of real complex networks by better identifying influential spreaders.

As a next step we recognized the deployment of the widely established h-index tool as a centrality metric, and introduced it's generalization in the domain of multilayer interconnected networks. Our work proposed a family of centrality metrics based on the h-index methodology of single networks. We take advantage of the multiple type of connections of a multilayer node, to efficiently detect such node entities that are strategically positioned in the multilayer network, e.g., accumulate a large number of connections from many (all) layers. All proposed methodologies are based on local knowledge of the network topology and are thus ideal for gigantic networks (e.g., Facebook, Twitter, LinkedIn) and real time applications. We evaluated the performance of our techniques for identifying influential spreaders in multilayer networks, that is, nodes that can rapidly spread information to as many layers as possible and as many nodes within each layer, respectively. We employed a wide range of competitors and their generalization in multilayer networks, e.g., PageRank, Betweenness, k-core etc. We found that the proposed method outperformed all competitors by providing a more accurate ranking for the spreading power of network nodes in real and semi-synthetic multilayer complex networks.

Next, we grasp the probabilistic nature of several networks in real life where connections are opportunistic. We thus focus on probabilistic complex networks where node connections are associated with weight values that may correspond to the mutual time spend by users of online social platforms or cost/gain of transition from one node to another, etc. We proposed a centrality metric that is based on limited length paths emanating from a node of interest (the focal node) and combine the weight values that correspond to those weighted interaction paths. We are thus interested in detecting probabilistic influential spreaders, that is, nodes that can efficiently disseminate information in weighted/probabilistic complex networks. We evaluated the proposed technique in a real network of student interactions, and several real complex networks where we assign the probabilistic link by following different probability distributions.

Following, we study spreading processes in the vehicular network. We focus on reducing redundant re-transmission in a network of vehicles, by selecting appropriate relay nodes, i.e., nodes that on behalf of the sender will further re-broadcast a message. We employ and appropriately modify a centrality metric from complex network theory and evaluate its performance with the optimized link state protocol (OLSR). The evaluation was conducted in a grid road network topology with a wealth of parameters regarding the communication range, vehicle velocity, acceleration etc. The proposed method outperformed its competitor by informing a significantly larger fraction of the vehicle nodes.

In the second part of this thesis, we follow a reverse policy and concentrate our efforts in blocking the outspread of undesired data (memes, rumors, viruses, etc.) in complex networks. We emphasize on the dynamic nature of spreading processes and try to address our objective while we follow the "virus" as it progresses through node communications. Most of the so far proposed techniques focus on static strategies (e.g., prior vaccination), however we believe that the problem is dynamic in nature and must be addressed appropriately. We proposed an algorithm that utilizes well studied heuristics from the literature of graphs−based on shortest paths−which was found to be quite effective in blocking the outspread of the diffusion. The evaluation over a wide range of real complex network and various simulation parameters, instructs that networks can be effectively protected by addressing the problem dynamically.

Next, we study malware propagation in vehicular networks. We propose a distributed solution for hindering the outspread of a virus by triggering a negating spreading process to counter the outspread of the malicious propagation. Inspired from complex network theory mechanisms we introduce two competing spreading process in the vehicular environment, where we try to shield vehicle nodes from malware propagated through vehicle communications. We utilize the dynamic nature of the vehicular network with aim to "outrun" the malicious diffusion by circulating among the vehicle nodes a list of infected (and potentially infected) vehicles and instruct healthy nodes to shut such communication paths. The evaluation was conducted via simulation in a real city map (Erlangen of Germany) extracted from openstreetmap. The simulation environment is composed of various intersection, building interfering with the communication and a wide range of vehicle and malware specific parameters. Our results illustrate that the proposed method can efficiently hinder the outspread of a virus until an appropriate patch arrives in the network, e.g., though cellular communication.

Following on our work in vehicular ad hoc networks we investigate on how routing protocols are affected by false data injected into the vehicular ecosystem by infected (with malware) vehicles. We employ several attack plans with aim to completely cancel out the benefits derived from vehicular communications. Particularly we inject fake measurements regarding CO_2 emissions and travel duration of specific road segments with aim to redirect vehicles to specific routes and create traffic congestion. Subsequently we employed a defense methodology that relies on vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and infrastructure-to-infrastructure (I2I) communications, that based on majority rules, successfully filters out fake data running through the systems communication phases, that is, restore the performance of our routing protocol to near normal operation.

The literature of social sciences is unimaginably rich and offers a wide range of findings from several disciplines. For our next part in this dissertation we focused on the *friendship paradox*, i.e., your friends have more friends than you do, and on how to utilize it (if possible) in complex networks. The paradox intuition is introduced because it contradicts people's common belief that they have more friends (on average) than their friends do. In real complex networks the notion of friendship is interpreted as connectivity, and it has been shown in the literature that it holds for the immediate connections (degree) of the network nodes. First we prove that the paradox holds not only for the degree centrality but also for a wide range of other centrality metrics as well (PageRank, Betweenness, Closeness, k-core, h-index, etc.). Additionally we provide solid proof that the paradox holds also for probabilistic characteristics such as the spreading power of the network nodes, that is, your immediate connections are more influential spreaders than you. Finally we examine the paradox paradigm for extended neighborhoods, i.e., two and three hop distant neighbors, and find that it strongly holds for the two hop neighbors as well. We evaluate our finding in the concept of sampling methods by selecting a random set of initial spreaders and evaluate their spreading (and blocking) potential when compared to that of their near neighbors (one, two and three hop vicinity). Our results on different real complex networks illustrate that the paradox intuition can straightforwardly be deployed to better identify super spreaders.

Finally, we take a glimpse of the Big Data ecosystem. The vast proliferation of networked devices has given rise to the era of data and colossal sized networks (in the number of nodes and connections) are emerging from continuous interactions of networked populations. The ever increasing magnitude of such network structures, the social networks, pose significant challenges to the industry and research communities. Hadoop has been widely deployed for

Big Data analysis, and the advent of solid state disks (SSDs) and their deployment in the Hadoop environment has been considered. In our work we empirical study the performance of solid state drives and hard disk drives for social network analysis, particularly in three directions; finding mutual friends among connected individuals; counting emergent triangle connection patterns; and finally calculating connected network components. These network/node characteristics have immediate effect on the spreading dynamics. Our work showed that the development of ”application profilers” that will try to predict the applications’ read/write pattern (random/sequential) and then incorporation of them into the Hadoop architecture will help reap the performance benefits of any current or new storage media.

ΠΕΡΙΛΗΨΗ

Διάχυση Πληροφορίας σε Σύμπλεκτα Δίκτυα με Εφαρμογές σε Δίκτυα Οχημάτων

από Παύλο Μπασαρά
Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
Πανεπιστήμιο Θεσσαλίας, Ελλάδα
Επίκουρος Καθηγητής Δημήτριος Κατσαρός, Πρόεδρος Επιτροπής

Με την άνευ προηγουμένου ανάπτυξη κατά την τελευταία δεκαετία διαφόρων τύπων κοινωνικών και επιχειρηματικών δικτύων, παράλληλα με τα φυσικά δίκτυα στις ανθρώπινες κοινότητες, η σύγχρονη κοινωνία βρίσκεται στα πρόθυρα της «πλήρης δικτύωσης». Οι συνεχόμενες εξελίξεις στις τεχνολογίες των πληροφοριών και των επικοινωνιών, σε συνδυασμό με τη δυνατότητα δημιουργίας και αποθήκευσης τεράστιου όγκου δεδομένων σχετικά με διάφορες πτυχές της ανθρώπινης (ηλεκτρονικής) συμπεριφοράς, κατέστησαν δυνατή την ανάλυση πολύπλοκων/πολυμερών δικτύων. Οι μελέτες κυμαίνονται από μεθοδολογίες βασισμένες στη θεωρία γράφων (μέγεθος και δύναμη των κοινοτήτων, ευρωστία σε επιθέσεις, μοντέλα ανάπτυξης δικτύων, συνδεσιμότητα κόμβου κ.ο.κ.), σε πιο κοινωνικο-θεωρητικές πτυχές (για παράδειγμα, ομοφυλία και διάδοση πληροφοριών). Αυτή η έρευνα έχει οδηγήσει στην υπολογιστική κοινωνική επιστήμη, ένα νέο τομέα που αξιοποιεί την ικανότητα συλλογής και ανάλυσης δεδομένων για την αποκάλυψη κρυφών μοτίβων σε μεμονωμένες και ομαδικές δραστηριότητες «συνδεδεμένων πληθυσμών». Τα αποτελέσματα της έρευνας αυτής στα ανερχόμενα πολύπλοκα δίκτυα βρίσκουν εφαρμογές σε πολλούς τομείς, συμπεριλαμβανομένου των τεχνολογιών μηχανών αναζήτησης [171] την ανάπτυξη **ad hoc** πρωτοκόλλων δικτύου [138] καθώς και την ανίχνευση και αναστολή της εξάπλωσης κακόβουλου υλικού [136]. Παράλληλα η έρευνα στα πολυμερή δίκτυα βρίσκει εφαρμογές και στα πλαίσια της ασφάλειας δικτύων, στα τρομοκρατικά δίκτυα [186] καθώς και στην διάδοση ιών. Παρόμοιες αναλύσεις συνήθως εφαρμόζουν τη θεωρία των γραφημάτων χρησιμοποιώντας μέτρα κεντρικότητας **(centrality measures)**, αλγόριθμους συντομότερης διαδρομής **(shortest paths)**, κατανομή και διασπορά της συνδεσιμότητας των κόμβων δικτύου **(degree distribution)**, και ούτω καθεξής.

Στην παρούσα διατριβή μελετάμε τα σύνθετα/πολυμερή δίκτυα από την οπτική γωνία της επιστήμης των δικτύων **(network science)** και της θεωρίας των γραφημάτων **(graph theory)**. Χρησιμοποιούμε εργαλεία, αλγορίθμους και μεθοδολογίες από την εκτεταμένη βιβλιογραφία της θεωρίας γραφημάτων, για να εμβαθύνουμε την γνώση μας στις ιδιότητες και τα διαφορετικά χαρακτηριστι-

κά των δικτύων (π.χ., την τοπολογία), δίνοντας έμφαση σε δυναμικές διεργασίες που λαμβάνουν χώρα στο εκάστοτε δίκτυο, όπως για παράδειγμα τη δυναμική διάδοση της πληροφορίας. Χρησιμοποιούμε μια σειρά από διαφορετικά εργαλεία (τα πιο ευρέως διαδεδομένα) για την πραγματοποίηση της έρευνας μας και την αξιολόγηση των προτεινόμενων τεχνικών, μηχανισμών και αλγορίθμων, ακολουθώντας ένα κοινό πλαίσιο μελέτης σε όλες τις φάσεις της έρευνας μας σε πραγματικά και τεχνητά δίκτυα.

Ξεκινάμε μελετώντας τα τοπολογικά χαρακτηριστικά των κόμβων δικτύου και το πως επηρεάζουν την ικανότητα του εκάστοτε κόμβου για την αποτελεσματική διάδοση πληροφοριών (διαφήμιση προϊόντων, ειδήσεων, κακόβουλου υλικού, κτλ.) πάνω από το δίκτυο. Κόμβοι που μπορούν να επηρεάσουν ένα μεγάλο πλήθος άλλων κόμβων (συγκρίσιμο με την τάξη μεγέθους του δικτύου) ονομάζονται κόμβοι σημαίνουσας επιρροής **(influential spreaders)** στο δίκτυο. Ένα μεγάλο μέρος της ερευνητικής κοινότητας εστιάζει στον σχεδιασμό τεχνικών και αλγορίθμων για την «εξόρυξη» σημαντικών κόμβων στα σύγχρονα δίκτυα. Κόμβοι σημαίνουσας επιρροής αναγνωρίζονται σε σχετικές μελέτες με βάση τη συνδεσιμότητα ενός κόμβου (βαθμός κόμβου - **degree)**, δηλ. οι κόμβοι που συγκεντρώνουν περισσότερες συνδέσεις θεωρούνται πιο σημαντικοί. Η μέθοδος κ-πυρήνα **(k-core decomposition)** είναι ακόμη ένα μέτρο κεντρικότητας που χρησιμοποιείται ευρέως για την ανεύρεση σημαντικών κόμβων στα σύγχρονα δίκτυα. Σχετικά αποτελέσματα αποδεικνύουν ότι η μέθοδος αυτή είναι πιο αποτελεσματική για την ανεύρεση κόμβων σημαίνουσας επιρροής σε σύγκριση με τη συνδεσιμότητα **(degree)** ενός κόμβου. Όμως η μέθοδος κ-πυρήνα απαιτεί σφαιρική/ολική γνώση της τοπολογίας του κάθε δικτύου, γεγονός που καθιστά την τεχνική ακατάλληλη για εφαρμογές σε πραγματικό χρόνο ή για δυναμικά μεταβαλλόμενα δίκτυα. Η έρευνα μας σε αυτόν τον τομέα εισήγαγε ένα νέο μέτρο κετνρικότητας, συγκεκριμένα το ***Power Community Index*** **(PCI)**, που βασίζεται αποκλειστικά σε τοπικά τοπολογικά χαρακτηριστικά (τοπική συνδεσιμότητα) του εκάστοτε κόμβου. Με βάση τις προαναφερθείσες ανταγωνιστικές τεχνικές, η μέθοδος **PCI** αποδείχθηκε ως η πιο αποτελεσματική μέθοδος για την εξόρυξη κόμβων σημαίνουσας επιρροής σε μια πληθώρα πραγματικών δίκτυων και διαφορετικών παραμέτρων για τη διάδοση της πληροφορίας.

Ως επόμενο βήμα εστιάζουμε στο μέτρο κεντρικότητας ***h*-index** (ευρέως διαδεδομένο μέτρο στα μεμονωμένα δίκτυα) και εισάγουμε τη γενικευμένη μορφή του στα πλαίσια των δικτύων με πολλαπλές συνδέσεις **(multilayer networks)**. Η δουλειά μας ανέπτυξε μια σειρά από μέτρα κεντρικότητας, που στον πυρήνα της χρησιμοποιεί την μεθοδολογία ***h*-index**. Εκμεταλλευόμαστε τον πολλαπλό τύπο συνδέσεων ενός κόμβου πολλαπλών επιπέδων **(multilayer node)**, για την αποτελεσματική ανίχνευση κόμβων που κατέχουν «στρατηγική» θέση στο δίκτυο, π.χ. συγκεντρώνουν μεγάλο αριθμό συνδέσεων από πολλά (όλα) επίπεδα. Οι προτεινόμενες μεθοδολογίες βασίζονται στην τοπική γνώση της τοπολογίας (συνδέσεων) του δικτύου και είναι επομένως ιδανικές για γιγαντιαία δίκτυα (π.χ. **Facebook, Twitter, LinkedIn)** και εφαρμογές σε πραγματικό χρόνο. Η αξιολόγηση των προτεινόμενων τεχνικών πραγματοποιήθηκε στα πλαίσια της ανεύρερης κόμβων σημαίνουσας επιρροής σε δίκτυα πολλαπλών συνδέσεων, δλδ. στην εξόρυξη κόμβων που μπορούν να διαδώσουν αποδοτικά πληροφορία σε όσο το δυνατόν περισσότερα επίπεδα καθώς και σε όσο το δυνατόν περισσότερους κόμβους σε κάθε επίπεδο ξεχωριστά. Χρησιμοποιούμε διάφορες ανταγωνιστικές μεθόδους που κατανοούν τα διαφορετικά χαρακτηριστικά των κόμβων με ξεχωριστό τρόπο όπως, π.χ., **PageRank, Betweenness, k-core** κλπ. Τα αποτελέσματα της έρευνάς μας (σε πραγματικά και συνθετικά πολυεπίπεδα δίκτυα) αναδεικνύουν την προτεινόμενη μεθοδολογία ως την πιο κατάλληλη τεχνική για τον πιο ακριβή διαχωρισμό/κατανομή των πολυεπίπεδων κόμβων με βάση την ικανότητα τους στην διάδοση πληροφορίας σε αυτό τον τύπο δικτύου.

Στη συνέχεια μελετούμε πολυμερή δίκτυα με πιθανολογικές συνδέσεις. Πιο συγκεκριμένα, εστιάζουμε σε πιθανοτικά δίκτυα όπου οι συνδέσεις κόμβων σχετίζονται με κάποια τιμή που αντιστοιχεί στο βάρος ακμής (σύνδεσης) και αντικατοπτρίζει για παράδειγμα τον αμοιβαίο/κοινό χρόνο που κόμβοι/χρήστες κοινωνικών δικτύων δαπανούν στα μέσα κοινωνικής δικτύωσης ή το κόστος/κέρδος μετάβασης από έναν κόμβο σε άλλο, και ούτω καθεξής. Σε αυτό τον τύπο δικτύου προτείνουμε ένα νέο μέτρο κεντρικότητας που βασίζεται σε μονοπάτια κόμβων περιορισμένου μήκους (με αρχή τον εκάστοτε κόμβο) συνδυάζοντας κατάλληλα τα βάρη ακμών που αντιστοιχούν στις συνδέσεις των μονοπατιών που προκύπτουν. Σε αυτό το κομμάτι της διατριβής εστιάζουμε στην ανεύρεση κόμβων σημαίνουσας επιρροής σε πιθανολογικά δίκτυα. Τα αποτελέσματα της έρευνας μας δείχνουν ότι το προτεινόμενο μέτρο κεντρικότητας, σε ένα σύνολο πραγματικών δικτύων με τεχνητά βάρη (διαφορετικών κατανομών) ανιχνεύει πιο αποτελεσματικά κόμβους σημαίνουσας επιρροής σε δίκτυα με πιθανολογικές συνδέσεις.

Στη συνέχεια της διατριβής, μελετάμε διαδικασίες διάχυσης πληροφορίας σε δίκτυα οχημάτων. Επικεντρωνόμαστε στο πρόβλημα της μείωσης της πλεονάζουσας αναμετάδοσης μηνυμάτων σε ένα δίκτυο οχημάτων, επιλέγοντας (με την χρήση μέτρων κεντρικότητας) στρατηγικά κατάλληλους κόμβους/οχήματα που θα έχουν το ρόλο του αναμεταδότη **(relay vehicle nodes)**, δηλ. κόμβους οι οποίοι αναλαμβάνουν την εκ νέου μετάδοση μηνυμάτων/πληροφορίας. Στόχος της έρευνας αποτελεί η διάδοση μηνυμάτων σε όσο το δυνατόν περισσότερα οχήματα στο οδικό δίκτυο. Χρησιμοποιούμε και κατάλληλα τροποποιούμε ένα μέτρο κεντρικότητας (συγκεκριμένα το **control centrality**) από τη θεωρία γράφων για την επιλογή κατάλληλων **relay** οχημάτων και αξιολογούμε την απόδοσή του σε σύγκριση με το βελτιστοποιημένο πρωτόκολλο **OLSR**. Η αξιολόγηση πραγματοποιήθηκε σε περιβάλλον προσομοίωσης χρησιμοποιώντας διαφορετικές παραμέτρους σχετικά με την εμβέλεια επικοινωνίας, την ταχύτητα και επιτάχυνση των οχημάτων, την τοπολογία του οδικού δικτύου, κ.α. Τα αποτελέσματα της έρευνας μας υποδεικνύουν ότι η προτεινόμενη τεχνική υπερτερεί έναντι της μεθόδου **OLSR** πετυχαίνοντας καλύτερη μετάδοση στο δίκτυο οχημάτων ενημερώνοντας μεγαλύτερο τμήμα (περισσότερα οχήματα) του δικτύου.

Στο δεύτερο μέρος της παρούσας διατριβής ακολουθούμε μια αντίστροφη πολιτική, επικεντρώνοντας την έρευνα μας σε σχετικές τεχνικές και αλγορίθμους από την θεωρία γράφων, έχοντας ως επίκεντρο μελέτης την παρεμπόδιση της διάχυσης πληροφορίας (κακόβουλες φήμες, ιούς, κτλ.) σε σύγχρονα/πολυμερή δίκτυα. Δίνουμε έμφαση στη δυναμική φύση της διαδικασίας μετάδοσης ακολουθώντας την διάδοση, π.χ. ενός ιού σε ένα δίκτυο επαφών **email**, καθώς ο ιός «προχωράει» μέσω των συνδεδεμένων επαφών. Οι περισσότερες μελέτες εστιάζουν σε στατικές στρατηγικές, ωστόσο πιστεύουμε ότι το πρόβλημα είναι δυναμικό/μεταβαλλόμενο και πρέπει να αντιμετωπιστεί κατάλληλα. Προτείνουμε μια μεθοδολογία που βασίζεται στην δυναμική ανεύρεση συντομότερων μονοπατιών σε κομμάτια δικτύου που χρήζουν άμεσης προσοχής, ακολουθώντας την μετάδοση βήμα προς βήμα. Η προτεινόμενη τεχνική ταξινομεί δυναμικά τις εκάστοτε ακμές/συνδέσεις και αφαιρεί (βάση περιορισμών) ένα μέρος αυτών, με στόχο την μείωση των πιθανών μονοπατιών προς «υγιείς» κόμβους. Η αξιολόγηση της προτεινόμενης τεχνικής περιλαμβάνει μια μεγάλη ποικιλία πραγματικών δικτύων και ανταγωνιστικών μεθόδων και αναδεικνύει την υπεροχή της, στην δυναμική παρεμπόδιση της διάδοσης κακόβουλου υλικού.

Στη συνέχεια, εξετάζουμε τη διάδοση ιών στα δίκτυα αυτοκινήτων. Συγκεκριμένα αναγνωρίζουμε την ανερχόμενη ανάγκη για την προστασία των σύγχρονων/μελλοντικών οχημάτων από ιούς και προτείνουμε μια κατανεμημένη προσέγγιση για την παρεμπόδιση της εξάπλωσης κακόβουλου λογισμικού μέσω των δυναμικών/ευκαιριακών συνδέσεων που προκύπτουν σε αυτό τον τύπο δικτύου. Η έρευνα μας έχει τις ρίζες της σε μηχανισμούς της θεωρίας γράφων όπου δύο ανταγωνιστικές διαδικασίες διάχυσης πληροφορίας−εξάπλωση του ιού & προτεινόμενη μέθοδος−ανταγωνίζονται πάνω από το δίκτυο για τον ίδιο πόρο, δηλαδή τους κόμβους/οχήματα. Η μέθοδος μας χρησιμοποιεί προς όφελος της την δυναμική φύση του δικτύου οχημάτων με σκοπό να ξεπεράσει την κακόβουλη διάχυση λογισμικού ενημερώνοντας τα «υγιή» οχήματα για την ύπαρξη των μέχρι στιγμής αναγνωρισμένων μολυσμένων (καθώς και πιθανός μολυσμένων) οχημάτων και αντίστοιχα το κλείσιμο της επικοινωνίας με τα εν λόγω οχήματα. Η αξιολόγηση της προτεινόμενης μεθοδολογίας πραγματοποιήθηκε σε περιβάλλον προσομοίωσης σε μια πραγματική τοπολογία δρόμου από την πόλη Ερλάνγκεν της Γερμανίας που αποτελείται από διάφορες διασταυρώσεις, δρόμους με διαφορετικές προτεραιότητες και εμπόδια (π.χ. κτίρια) που παρεμβαίνουν στην επικοινωνία των οχημάτων. Τα αποτελέσματα της έρευνας μας δείχνουν ότι η προτεινόμενη μέθοδος μπορεί αποτελεσματικά να μειώσει την εξάπλωση του κακόβουλου λογισμικού στο δίκτυο έως ότου ένα κατάλληλο «φάρμακο» διανεμηθεί στα οχήματα π.χ. μέσω κυψελοειδούς επικοινωνίας.

Σε συνέχεια της έρευνας μας στα **ad hoc** δίκτυα οχημάτων εστιάζουμε στην επίδραση της ψευδούς πληροφορίας σε πρωτόκολλα αναδρομολόγησης. Υποθέτουμε την ύπαρξη μολυσμένων (με κακόβουλο λογισμικό) οχημάτων που εισάγουν στο σύστημα ψευδή πληροφορία για την κυκλοφοριακή κατάσταση στην τοπολογία δρόμου που μελετούμε, και εξετάζουμε το πως αυτό επηρεάζει τις αποφάσεις αναδρομολόγησης στο πρωτόκολλο. Αρχικά χρησιμοποιούμε διάφορα σχέδια επίθεσης με σκοπό να καταργήσουμε εντελώς τα οφέλη που παρέχουν οι επικοινωνίες οχημάτων. Συγκεκριμένα εισάγουμε ψευδείς μετρήσεις στο σύστημα ακολουθώντας διαφορετικές μεθοδολογίες, σχετικά με τις εκπομπές διοξειδίου του άνθρακα καθώς και την διάρκεια διάσχισης συγκεκριμένων τμημάτων δρόμου. Απώτερος σκοπός μας είναι η αναδρομολόγηση των οχημάτων σε συγκεκριμένα σημεία της τοπολογίας δρόμου και η δημιουργία κυκλοφοριακής συμφόρησης. Στην συνέχεια προτείνουμε ένα μηχανισμό άμυνας ο οποίος βασίζεται στην επικοινωνία οχήματος με οχήμα, οχήματος-υποδομής και επικοινωνίες υποδομής με υποδομή, χρησιμοποιώντας κατάλληλους κανόνες για τον εντοπισμό των ψευδών δεδομένων που έχουν εισαχθεί στο σύστημα από μολυσμένα οχήματα. Η προτεινόμενη μεθοδολογία αποδείχθηκε ικανή στον να «αντιληφθεί» τα ψευδή δεδομένα και στην επαναφορά του του πρωτοκόλλου δρομολόγησης σε σχεδόν κανονική λειτουργία.

Η βιβλιογραφία των κοινωνικών επιστημών προσφέρει ένα εκτενές φάσμα ευρημάτων που βρίσκει ανταπόκριση σε διάφορους επιστημονικούς κλάδους. Στο επόμενο μέρος της παρούσας διατριβής εστιάζουμε στο παράδοξο της φιλίας **(friendship paradox)**: οι φίλοι σου έχουν (κατά μέσο όρο) περισσότερους φίλους από εσένα, και στο πως αυτό το κοινωνικό φαινόμενο μπορεί να χρησιμοποιηθεί στα πολύπλοκα δίκτυα. Η έννοια του «παράδοξου» έγκειται στο γεγονός ότι αντιφάσκει την κοινή πεποίθηση των ανθρώπων ότι έχουν περισσότερους φίλους από τους φίλους τους. Στα σύγχρονα δίκτυα η έννοια της «φιλίας» ερμηνεύεται ως συνδεσιμότητα-επικοινωνία (ακμή), και έχει αποδειχθεί ότι το παράδοξο ισχύει στα πολυμερή δίκτυα για το βαθμό συνδεσιμότητας **(degree)** των κόμβων. Σε συνέχεια της έρευνας αυτής αποδεικνύουμε ότι το «παράδοξο» ισχύει ακόμη για μια σειρά από μέτρα κεντρικότητας που δεν σχετίζονται άμεσα με το βαθμό συνδεσιμότητας των κόμβων όπως για παράδειγμα η μέθοδος **PageRank, Betweenness, Closeness, k-core, *h*-index** κ.α. Επιπλέον αποδεικνύουμε ότι το παράδοξο ισχύει ακόμη και σε πιθανολογικά χαρακτηριστικά όπως η δύναμη επιρροής **(power of influence)** των κόμβων στο δίκτυο, με άλλα λόγια, οι άμεσες συνδέσεις σου είναι κόμβοι μεγαλύτερης σημαίνουσας επιρροής από εσένα. Τέλος, εξετάζουμε το παράδοξο όχι μόνο σε σχέση με τις άμεσες συνδέσεις του εκάστοτε κόμβου αλλά επιπρόσθετα και για πιο μακρινούς γείτονες **(2-3 hop)** και επιβεβαιώνουμε την ισχύ του παράδοξου και σε αυτές τις περιπτώσεις. Τα αποτελέσματα της έρευνάς μας μπορούν να χρησιμοποιηθούν άμεσα σε μεθόδους τυχαίας δειγματοληψίας σημαντικών κόμβων σε γιγαντιαία δίκτυα καθώς και στην σημαντική βελτίωση αλγορίθμων για την ανεύρεση κόμβων σημαίνουσας επιρροής σε σύγχρονα δίκτυα.

Τέλος εστιάζουμε στην επιστήμη των **Big Data**. Ο συνεχόμενος πολλαπλασιασμός των συσκευών με δυνατότητες επικοινωνίας στην καθημερινή μας ζωή οδήγησε στην σύγχρονη εποχή των δεδομένων και των δικτύων γιγαντιαίων διαστάσεων (στον αριθμό των κόμβων και των συνδέσεων). Συνεπώς η βιομηχανία και οι ερευνητικές κοινότητες βρίσκονται συνεχώς αντιμέτωποι με νέες προκλήσεις. Το περιβάλλον **Hadoop** έχει αναπτυχθεί ευρέως για την ανάλυση των **Big Data** και η χρήση των αποθηκευτικών «δίσκων στερεάς κατάστασης» **(solid state discs)** είναι πολλά υποσχόμενη για εφαρμογή στο κατανεμημένο σύστημα αρχείων του **Hadoop**. Στην έρευνα μας εξετάζουμε την απόδοση των μονάδων στερεάς κατάστασης σε σύγκριση με παραδοσιακούς σκληρούς δίσκους για την ανάλυση σύγχρονων κοινωνικών δικτύων. Συγκεκριμένα εστιάζουμε σε τρεις κατευθύνσεις: (α) στην ανεύρεση κοινών φίλων μεταξύ συνδεδεμένων χρηστών, (β) στην καταμέτρηση συνδεσμολογιών τριγώνου καθώς και (γ) στον υπολογισμό των συνδεδεμένων μερών **(connected components)** των δικτύων. Τα παραπάνω χαρακτηριστικά συνδέονται άμεσα με την αποδοτική διάδοση της πληροφορίας στα σύγχρονα κοινωνικά δίκτυα. Η έρευνα μας αναδεικνύει την ανάγκη για δημιουργία προφίλ εφαρμογών, που θα προσπαθήσει να προβλέψει το πρότυπο ανάγνωσης/εγγραφής της εκάστοτε εφαρμογής (τυχαία/διαδοχικά), και θα βοηθήσει τη αρχιτεκτονική **Hadoop** στον να αποκομίσει τα πλεονεκτήματα απόδοσης οποιουδήποτε τρέχοντος ή νέου μέσου αποθήκευσης.

PUBLICATIONS

Submitted Journals

[S1.] Pavlos Basaras, Giorgos Iosifidis, Dimitrios Katsaros, Leandros Tassiulas. *On neighboring nodes' relative power of influence*, **Submitted for journal publication**, October 2017.

Articles in Journlas

[J1.] Pavlos Basaras, Giorgos Iosifidis, Dimitrios Katsaros, Leandros Tassiulas. *Identifying Influential Spreaders in Complex Multilayer Networks: A centrality perspective*, **IEEE Transactions on Network Science and Engineering**, accepted, October, 2017.

[J2.] Marios Bakratsas, Pavlos Basaras, Dimitrios Katsaros, Leandros Tassiulas. *Hadoop MapReduce performance on SSDs for analyzing social networks*, **Big Data Research (Elsevier)**, accepted, June, 2017.

[J3.] Pavlos Basaras, Dimitrios Katsaros, Leandros Tassiulas. *Detecting Influential Spreaders in Complex, Dynamic Networks*, **IEEE Computer magazine**, vol. 46, no. 4, pp. 26-31, April, 2013.

Articles in Conference Proceedings

[C1.] Marios Bakratsas, Pavlos Basaras, Dimitrios Katsaros, Leandros Tassiulas. *Hadoop MapReduce performance on SSDs: The case of complex network analysis tasks*, **Proceedings of the 2nd Neural Network Society International Conference on BigData (INNS BigData)**, vol. 529, pp. 111-119, Thessaloniki, Greece, October 23-25, 2016.

[C2.] Pavlos Basaras, Ioannis-Prodromos Belikaidis, Leandros Maglaras, Dimitrios Katsaros. *Blocking Epidemics Propagation in Vehicular Networks*, **Proceedings of the 12th IEEE/IFIP Annual Conference on Wireless On-demand Network Systems and Services (WONS)**, pp. 65-72, Cortina d'Ampezzo, Italy, January 20-22, 2016.

[C3.] Pavlos Basaras, Leandros Maglaras, Dimitrios Katsaros, Helge Janicke. *A Robust Eco-Routing Protocol Against Malicious Data in Vehicular Networks*, **Proceedings of the 8th IFIP Wireless and Mobile Networking Conference (WMNC)**, pp. 184-191, Munich, Germany, October 5-7, 2015.

[C4.] Pavlos Basaras, Dimitrios Katsaros, Leandros Tassiulas. *Dynamically Blocking Contagions in Complex Networks by Cutting Vital Connections*, **Proceedings of the IEEE International Conference on Communications (IEEE ICC)**, pp. 1170-1175, London, UK, June 8-12, 2015.

[C5.] Alexandra Stagkopoulou, Pavlos Basaras, Dimitrios Katsaros. *A Social-based Approach for Message Dissemination in Vehicular Ad Hoc Networks*, **Proceedings of the 6th International Conference on Ad Hoc Networks**, vol. 140, Springer, pp. 27-38, Rhodes island, Greece, August 18-19, 2014.

Chapters in Books

[B1.] Pavlos Basaras, Dimitrios Katsaros. *Identifying Influential Spreaders in Complex Networks with Probabilistic Links*, **In (Tansel Ozyer, ed.) Social Network and Surveillance for Society**, chapter in book, Springer, accepted, September, 2017

[B2.] Dimitrios Katsaros, Pavlos Basaras. *Detecting Influential Nodes in Complex Networks with Range Probabilistic Control Centrality*, **chapter in Coordination Control of Distributed Systems (Jan H. van Schuppen and Tiziano Villa)**, Lecture Notes in Control and Information Sciences, vol. 456, Springer-Verlag, pp. 265-272, 2015.

In addition, our research efforts within the same period led to the following publications that are not directly related to this thesis:

Articles in Conference Proceedings

[M1.] Dimitrios Papakostas, Pavlos Basaras, Dimitrios Katsaros, Leandros Tassiulas. *Backbone Formation in Military Multi-Layer Ad Hoc Networks Using Complex Network Concepts*, **Proceedings of the 35th IEEE Military Communications Conference (MILCOM)**, pp. 842-848, Baltimore, Maryland, USA, November 1-3, 2016.

[M2.] Nikos Makris, Pavlos Basaras, Thanasis Korakis, Navid Nikaein, Leandros Tassiulas. *Experimental Evaluation of Functional Splits for 5G Cloud-RANs*, **Proceedings of the IEEE International Conference on Communications (IEEE ICC)**, Paris, May 21-25, 2017.

[M3.] Umer Khan, Pavlos Basaras, Lars Schmidt-Thieme, Alexandros Nanopoulos, Dimitrios Katsaros. *Analyzing Cooperative Lane Change Models for Connected Vehicles*, **Proceedings of the 3rd International Conference on Connected Vehicles and Expo (ICCVE)**, pp. 565-570, Vienna, Austria, November 3-7, 2014.

[M4.] Leandros Maglaras, Pavlos Basaras, Dimitrios Katsaros. *Exploiting Vehicular Communications for Reducing CO2 Emissions in Urban environments*, **Proceedings of the 2nd International Conference on Connected Vehicles and Expo (ICCVE)**, pp. 32-37, Las Vegas, Nevada, USA, December 2-6, 2013.

Acknowledgements

This thesis represents the research conducted towards completing my PhD Degree from the Department of Electrical & Computer Engineering, University of Thessaly, Greece. The conclusion of this PhD study looks like the end of a long journey comprised of strong emotions of patience, persistence and joy. First and foremost I want to thank my supervisor *Assistant Prof. Dimitrios Katsaros* for guiding me through this journey, supporting me all those years with his advice and guidance day by day. Through his experience I gained valuable lessons that will accompany me in my life and career henceforth. I have no words to express my gratitude to him.

I am particularly indebted to *Prof. Leandros Tassiulas* for his guidance, support and motivation during my research, and for giving me the opportunity to work in such an inspiring research team. I express to *Prof. Leandros Tassiulas* and *Assistant Prof. Thanasis Korakis* my sincere gratitude for their support all those years, for giving me the opportunity to participate in European projects and broaden my knowledge in many aspects. To *Ussher Assistant Prof. Giorgos Iosifidis* I want to express my sincere thanks for supporting me in several phases of my PhD work. His fruitful criticism and continuous support always improved the quality of my research directions and solutions. I am also grateful to *Lecturer Leandros Maglaras* for supporting me at the initial steps of my PhD. I am very gratefull to *Manolopoulos Yannis* and *Tsoukalas Lefteris* for their valuable comments and support for my dissertation. To *Associate Prof. Spyros Lalis* and *Associate Prof. Apostolos Papadopoulos* I want to express my sincere thanks for accepting to serve in the examination committee of my thesis and especially *Associate Prof. Apostolos Papadopoulos* for his comments.

Moreover, I would like to thank all my lab-mates. Particular thanks to the NITLab team: *Harris Niavis, Giannis Kazdaridis, Nikos Makris, Donatos Stavropoulos, Kostas Choumas, Stratos Keranidis, Xristos Zarafetas, Ilias Syrigos, Virgilios Passas, Kostas Chounos, Dimitris Giatsios, Giannis Igoumenos, Apostolos Apostolaras, Aris Dadoukis, Antonis Kalkanof, Vasilis Miliotis, Panagiotis Skrimponis, Giannis Zografopoulos, Christina Madelou...* Their support all those years have shaped my personality so far.

I want to express my most sincere thanks and appreciation to my closest friends *Spyros Konstantis* and *Grammatiki Papagianni*, for being at my side for more than fifteen years now, always encouraging me and supporting me in every aspect of my life. I have no words to express my feelings for them for their presence in my life.

Last but not least, I want to thank my family, my mother *Dimitra*, my brother *Giorgos* and my father *Vasilis*. I have no words to express my gratitude for them, their support and love for all my decisions. I dedicate this dissertation to them.

Table of Contents

Page

List of Tables **xxv**

List of Figures **xxvii**

I Introduction **1**

1 Introduction **3**

1.1 Motivation 3

1.2 Synopsis 9

II Spreading Dynamics in Complex, Multilayer and Vehicular Networks **13**

2 Accelerating Spreading Processes in Single Complex networks **15**

2.1 Introduction 15

2.1.1 Motivation 16

2.2 Identifying Influential Spreaders 16

2.3 Balancing Betweenness and Coreness 17

2.4 Performance Evaluation 17

2.4.1 Single original spreader 19

2.4.2 Multiple original spreader 21

2.5 Conclusion 22

3 Accelerating Spreading Processes in Probabilistic Complex Networks **23**

3.1 Introduction 23

3.1.1 Motivation and contributions 24

3.2 Related Work 25

3.3 Proposed Technique 26

3.3.1 Complex Networks with Probabilistic Links 27

3.3.2 r-Hop User Communication Paths (UCPs) 27

3.3.3 Range Probabilistic Communication Area (rPCA) 28
3.4 Performance evaluation 29
3.4.1 Competing Techniques 29
3.4.2 Simulation Settings 31
3.4.3 Propagation Model and Influence 32
3.4.4 Evaluation Criteria 32
3.5 Results 33
3.5.1 Impact of infection probability 33
3.5.2 Impact of Zipfian skewness 38
3.5.3 Evaluation with a real complex network 39
3.6 Conclusions 41

4 Accelerating Spreading Processes in Multilayer Complex Networks 43
4.1 Introduction 43
4.2 Preliminaries 45
4.2.1 Monoplex, multiplex and multilayer networks 46
4.2.2 Diffusion in multilayer networks 46
4.3 Proposed methods to identify highly influential spreaders 47
4.3.1 The family of multilayer PCI measures 48
4.4 Evaluation settings 50
4.4.1 Competitors for multiplex networks 50
4.4.2 Competitors for multilayer networks 51
4.4.3 Summary of competitors 51
4.4.4 Datasets 51
4.4.5 How to evaluate the performance 54
4.4.6 Setting parameters 55
4.5 Results 55
4.5.1 Ranking influence in real networks 55
4.5.2 Ranking influence in semi-synthetic networks 60
4.6 Conclusion 65

5 Accelerating Spreading Processes in Vehicular Networks 71
5.1 Introduction 71
5.2 Control Centrality 72
5.2.1 From Control Centrality to pCoCe 73
5.3 Relay selection 75
5.3.1 Selecting relays through pCoCe 75
5.3.2 Selecting relays through OLSR 75
5.4 Performance Evaluation 75

5.4.1 Simulation design 76
5.5 Results 76
5.5.1 Experimenting on vehicle density, 2pCoCe 76
5.5.2 Differences in the selected relays 77
5.5.3 Increasing the range of pCoCe to 3 hops distance 78
5.5.4 Reducing the range of communication to 250m 78
5.6 Conclusion 80

III Blocking the Outspread of Undesired Data in Complex and Vehicular Networks 81

6 Blocking the Outspread of Undesired Data in Complex Networks 83
6.1 Introduction 83
6.2 Problem Formulation 85
6.3 Critical Edge Detector (CED) 85
6.4 Performance Evaluation 87
6.4.1 Datasets 87
6.4.2 Simulation Design 87
6.4.3 Competing Methods 89
6.4.4 Results 89
6.5 Conclusion 94

7 Blocking the Outspread of Undesired Data in Vehicular Networks 95
7.1 Introduction 95
7.1.1 Motivation and contributions 96
7.2 Related work 97
7.3 Virus Propagation 98
7.4 Proposed Mechanism 98
7.4.1 Specialized Hardware (SH) 98
7.4.2 Isolating Infectious Vehicles 99
7.5 Experimental Design 101
7.5.1 Simulators 101
7.5.2 Map 101
7.5.3 Initially Infected Vehicles 103
7.5.4 Vehicle Settings 103
7.6 Results 104
7.6.1 Impact of Vehicle Density & Different Initial Spreader 105
7.6.2 Impact of Infection Delay (τ) 106
7.6.3 Impact of Virus Strength 106

7.6.4 Impact of Different Cut Methods 107
7.7 Conclusion 108

8 Protecting a Vehicular Network from Infected Nodes 111
8.1 Introduction 111
8.2 Related Work 112
8.3 Preliminary Work, $ErouVe$ 113
8.3.1 System Description 114
8.3.2 System Initialization 114
8.3.3 Communication Phases 115
8.3.4 New Decision System for Route Selection 115
8.4 ErouVe Vulnerabilities 116
8.5 Attack Plans 117
8.5.1 Attack Objectives 117
8.5.2 How To Attack 118
8.6 Proposed Defense System: Enhanced ErouVe 118
8.6.1 Fake Route Countermeasures 119
8.6.2 Fake Data Countermeasures 119
8.7 Simulation Settings 120
8.7.1 Simulator 120
8.7.2 Evaluation Scenario 121
8.7.3 Communication Settings 121
8.7.4 Parameters 122
8.8 Performance Evaluation 122
8.8.1 ErouVe VS Shortest Path VS FR attacks 122
8.8.2 Impact of Attack Group Size 123
8.8.3 Impact of Attack Interval 123
8.8.4 Impact of Defense System VS FD attacks 125
8.9 Conclusion 125

IV Low Cost Sampling Methodologies Based on Social Driven Aspects 127

9 On neighboring nodes' relative power of influence 129
9.1 The influence power of my close neighbors 130
9.2 Results 132
9.2.1 The centrality paradox 133
9.2.2 The spreading paradox 135
9.3 Applications 138
9.3.1 Mining Cascade Initiators/Blockers 138

9.3.2 Accelerating the Spreading Process 138
9.3.3 Blocking the Outspread of Misinformation 139
9.4 Discussion 143
9.5 Materials and Methods 144
9.5.1 Data description 144
9.5.2 Individual and network level property 145

V Implementation Issues on the Hadoop Environment 147

10 Hadoop MapReduce performance on SSDs 149
10.1 Introduction 149
10.2 Related work 151
10.3 Hadoop structure 152
10.4 Investigated algorithms 153
10.4.1 Mutual friends 154
10.4.2 Connected components 156
10.4.3 Counting triangles 156
10.5 Experimental environment and results 158
10.5.1 System setup 158
10.5.2 Input data and performance measures 159
10.5.3 Results 160
10.6 Conclusions 166

11 Conclusions & Future Work 169

A Materials and Methods 173
A.1 Spreading models 173
A.1.1 Susceptible-Infectious-Recovered (SIR) 173
A.1.2 Susceptible-Infectious-Susceptible (SIS) 174
A.2 Centrality Metrics 175
A.3 Performance Evaluation 176

B Appendix of Chapter 4 179
B.1 Multilayer network generator 179

C Supplementary for “On neighboring nodes’ relative power of influence" 185
C.0.1 Detailed experiments on the centrality paradox at the network level 185
C.0.2 Detailed experiments on the centrality paradox at the individual level . . 186
C.0.3 Detailed experiments for the blocking application under the SIR model . . 195
C.0.4 Detailed experiments regarding the spreading application for the SIR model196

C.0.5 Detailed experiments for the spreading application of the SIS model . . . 202
C.0.6 Detailed experiments on the spreading paradox at the individual level: SIR spreading model 210
C.0.7 Detailed experiments on the spreading paradox at the individual level: SIS spreading model 211

Bibliography **215**

List of Tables

Table **Page**

2.1 Complex Network Attributes 18
2.2 Number of influential spreaders that can maximize infection in three networks. . . . 21

3.1 Networks base attributes. 31

4.1 Notation for multilayer networks. 46
4.2 A summary of competing methods evaluated. 52
4.3 Multiplex networks. 52
4.4 Layers of semi-synthetic networks. 53
4.5 Stability of ranking with respect to the average spreading power. The values represent the ratio between the correlation (τ) of a competitor, and the best performing method (i.e., $mlPCI$). 54
4.6 Experimentation parameters. 55

6.1 Network Base Attributes 87

7.1 Simulation Parameters 104

8.1 Example of Connections Table for 3 RSUs 115
8.2 Simulation Parameters 122

9.1 Fraction of nodes that the paradox holds at the individual level. 144
9.2 Characteristics of examined complex networks. Apart from the number of nodes and edges, the table also depicts the epidemic threshold (ϵ), the average degree (k), and the type of the network. 145

10.1 Characterization of problems/algorithms examined. 154
10.2 Computer specifications. 159
10.3 Installed software. 159
10.4 Custom settings. 159
10.5 Social networks used for evaluation. 160
10.6 Average times for each phase for 2nd job (creating triples) of “mutual friends” algorithm.162

10.7 Average times for each phase for 1st job (forming triads) of “counting triangles” algorithm. 162

10.8 Average times for each phase for 2nd job (counting triangles) of “counting triangles” algorithm. 162

10.9 Average times for each phase for 1st job (create triads) of “counting triangles” algorithm, with changed container’s settings. 163

10.10 Performance difference for YouTube dataset at “Counting Triangles”, increasing sort factor, for HDD. 163

10.11 Performance difference for YouTube dataset at “Counting Triangles”, increasing sort factor, for SSD2. 163

10.12 Performance difference for YouTube dataset at “Counting Triangles”, increasing file buffer size, for HDD. 164

10.13 Performance difference for YouTube dataset at “Counting Triangles”, increasing file buffer size, for SSD2. 164

10.14 Percentage difference between “customs” and “containers” settings for YouTube dataset, at “Counting Triangles” algorithm. 164

10.15 Percentage difference between “customs” and “containers” settings for YouTube dataset, at “Mutual Friends” algorithm. 164

10.16 Sum of average times for each phase for the iterative jobs of “Connected Components”. 165

List of Figures

Figure **Page**

1.1 Multiplex network of European airlines. 4

1.2 A network of vehicles and road side units. 5

1.3 Transition probabilities between the different states of various spreading models. . . 8

2.1 Spreading capability of nodes in the ca-CondMat network with a single original spreader according to (a) 1-PCI and (b) k-shell index. There are nodes with high k-shell indices, some of which infect a large portion of the network, as well as nodes with the same k-shell index (16) that infect a significantly smaller part of the network. On the other hand, only nodes with very small 1-PCI exhibit such behavior. 19

2.2 Spreading capability of nodes in the CA-AstroPh network with a single original spreader according to (a) 1-PCI and (b) k-shell index versus node degree. The k-shell index fails to fulfill monotonicity in many cases, and 1-PCI has a better correlation with node degree. 20

2.3 Spreading capability of nodes in the ca-AstroPh network with multiple original spreaders according to node degree, 1-PCI, and k-shell index. The k-shell index is the least effective measure. Node degree is the most effective measure, closely followed by 1-PCI, but the discrepancy between these values quickly diminishes as the number of multiple original spreaders grows. 21

3.1 $rPCA$ identifies nodes which possess the characteristic that from these nodes emanate "strong" paths. For 2 hops distance: $2PCA(a) = 17.283$ and $2PCA(b) = 1.1$ assuming that both i and j have 2 outgoing neighbors and x, z are hypothetic nodes, i.e., not included. 27

3.2 Ranked percent with respect to the total number of nodes of each network case for all evaluated λ values, i.e., nodes with $IF > 0$. 33

3.3 In almost all different spreading rates for the ego-Twitter network, the proposed technique significantly outperforms its competitors. 34

3.4 For the soc-Slashdot0922 network we observe that our approach coincides with the rest of the competing algorithms only for the higher spreading rates. 34

3.5 As the spreading rate increases, our two-fold approach maintains its superior performance as compared to the rest of the competing techniques. 35

3.6 For the final network case, an oscillation for the most accurate ranking is observed at the lower spreading rates. Nonetheless, the proposed technique is found within the higher τ values. 36

3.7 $wClo$ was found to coincide with the proposed technique in a few configurations. The presented heat plots, illustrate that influence is closer related with 2PCA. On the contrary, for $wClo$, we observe that the medium values depict an amplitude of influence values. 38

3.8 Ranging in skewness for the distribution of links. The spreading rate is set at 2%. . . 39

3.9 Distributions of the link weight (i.e., aggregated contact duration) of the real weighted network. 40

3.10 Evaluation of competing algorithms over the real weighted network. 41

4.1 A multilayer network consisting of four layers L1, L2, L3 and L4. Nodes with the same ID in different layers depict clones of the same node. 47

4.2 Rankings capabilities (Kendall's Tau b) of all competing techniques in real multiplex networks with respect to λ_{ii}. It can be observed that all competing algorithms exhibit similar trends, i.e., either increasing or decreasing trend as the intra-spreading probability changes. $mlPCI$ illustrates the largest correlation with influence in almost all networks. While $mlPCI$ shows a relatively stable behavior, i.e., it is (almost) always at the top of the ranking chain, the remaining algorithms do not posses that property as their rank changes in the different networks, e.g., $aggDeg$ is 2nd in Homo and 6th in MoscowAthletics2013. 56

4.3 Distribution of $alPCI$ values for all networks. It can be observed that for most networks the majority of nodes has relatively low $alPCI$ values, whereas the largest indexes are appointed to only a few nodes. 57

4.4 Distribution of $sumCore$ values for all networks. According to the illustrated distributions, we observe two groups: (Drosophila, MoscowAthletics2013, NYClimateMarch) and (Homo, Sacchpomb, Sacchcere). 58

4.5 Maximum cascade size per layer subject to the distribution of interconnections. It can be observed that when all parameters are set to 0.3 the cascade size is maximum, while the opposite occurs, when all parameters are set to 0.8. 61

4.6 Rankings capabilities (Kendall's Tau b) of all competing techniques in real networks with synthesized interconnections with respect to uncorrelated with influence in these networks, because it assigns to almost all network nodes the same index value. 67

4.7 Rankings capabilities (Kendall's Tau b) of all competing techniques in real networks with synthesized interconnections with respect to λ_{ij}. $mlPCI$ remains at the top of the ranking chain. $verPR$'s performance is better in the SLN networks where interconnections are more dense (when compared to the intra-connections) with respect to the DLN networks, and particularly is at its best when s_{node} or s_{layer} is 0.8. It can be observed that measuring the influence capabilities of a node by counting the number of geodesics that pass through that node ($aggBC$, $verBC$) does not yield competitive results. 68

4.8 Increasing in the number of interconnections in the SLN networks. It can be observed that all methods illustrate a decreasing trend as d increases. Setting s_{node} at 0.8 and thus assigning to a specific set of nodes many interconnections, works in favor of $verPR$ which exhibits an exceptional performance in this case. 69

4.9 Increasing in the number of interconnections in the DLN networks. As interconnections increase $alPCI$ yields better results, i.e., from 4th when $d = 1$ to 1st when $d = 4$. It's performance is different from the SLN networks because for the DLN networks, the distribution of inter-k_{out} is still significantly lower (even for $d = 4$) from that of intra-k_{out} (compare Figures B.1 and B.2 with Figure B.4 in the Appendix) which does not hold for the SLN networks. 69

5.1 Illustration of a stem-cycle disjoint subgraph. 73

5.2 The out-neighbors of vehicle S are illustrated. 74

5.3 Link quality between vehicle nodes. 74

5.4 OLSR Vs 2pCoCe at different velocities for sparse and dense scenarios. 77

5.5 Normalizing the coverage ratio of each method with respect to the average number of selected relays. 78

5.6 Comparing pCoCe's performance with 2 and 3 hops distance. 79

5.7 Communication range at 250m for frequency of vehicles every 1 seconds. 79

6.1 Generalized framework for blocking epidemic outbreaks in Complex Networks. This article focuses on dynamic strategies and edge removing mechanisms to hinder the spread of misinformation. 84

6.2 In the current time step (t) the infected nodes are assumed to be 'a' and 'm' whereas n1 and n2 are the infected sources of the immediate previous step (t-1) which are now immunized (removed). The dashed lines correspond to the three hop abstract network images, as seen from the perspective of the current infected sources. 86

6.3 The strength of the propagation is 6%. The initially infected set is connected to the immediate vicinity with 548 connections whereas the lost fraction of nodes for the unblocked diffusion is about 280 nodes. As we increase in the x-axis CED's better performance becomes more evident. 90

6.4 The strength of the propagation is 4%. The initially infected set is connected to the immediate vicinity with 410 connections whereas the lost fraction of nodes for the unblocked diffusion is about 360 nodes. For this weakly connected network all methods illustrate a good performance. 90

6.5 The strength of the propagation is 6%. The initially infected set is connected to the immediate vicinity with 3400 connections whereas the lost fraction of nodes for the unblocked diffusion is about 1270 nodes. Only the proposed technique manages to hinder the propagation sufficiently in the later steps of β. 91

6.6 The strength of the propagation is 2%. The initially infected set is connected to the immediate vicinity with 11285 connections whereas the lost fraction of nodes for the unblocked diffusion is about 2080 nodes. Again the network is better protected by CED. 91

6.7 The y-axis represents the fraction of saved nodes with regard to the lost nodes of the unblocked diffusion (814, 1048, 1270, 1488, 1714) respectively. Our approach seems to be affected by the increase of λ significantly later than its competitors. 92

6.8 The y-axis represents the fraction of saved nodes with regard to the lost nodes of the unblocked diffusion (113, 190, 280, 385, 511) respectively. CED illustrates better results by securing a significantly larger part of the network's interacting nodes for all λ values. 93

7.1 Vehicle B is informed of A's infection by the SH. B will further broadcast (and exchange) its version of the BL with all other vehicles found in its trajectory. 100

7.2 Part of the Erlangen city. SHs are positioned near the center of the map. The illustrated scanning region is indicative, to highlight the relatively short range of the specialized hardware devices. 102

7.3 Percentage of the infected network from the different initial spreading points. 105

7.4 Average infected network size. 106

7.5 Impact of the tranmissibilty of the virus. 107

7.6 Vulnerability of vehicles to infection. 107

7.7 Cutting different neighbors from infected nodes. 108

8.1 CO_2 emissions reduction system based on DSRC communications 114

8.2 New decision mechanism 116

8.3 Fake Data Countermeasures 120

8.4 Simulation Map 121

8.5 FR successfully deceives the original algorithm into sending vehicles to the short route and thus creating congestion. Travel duration and CO_2 emissions are significantly increased by 31% and 20% respectively. 123

8.6 As the number of FD attacks running in system increases, ErouVe's performance drops. About 30% of vehicles out of the total simulation were bogus (attack group size set to 5) for a 25% decrement in travel duration. 124

8.7 In order to significantly affect the routing decisions of ErouVe, fake data need to arrive in a timely manner, so as to continuously have false data in the system. Otherwise ErouVe may quickly recover to original routing instructions. 124

8.8 The proposed defense system returns the protocol to near identical routing decisions by successfully filtering out the outliers and thus the overall system's performance is preserved. 125

9.1 Centrality paradox at network level. The x-axis shows the evaluated centralities measures while the y-axis illustrates the distance in ratio $1-\frac{\langle v\rangle}{\langle v_{nn}\rangle}$ for all neighborhoods (N_1, N_2 and N_3). The paradox holds for networks with power-law degree distribution due to the existence of hub nodes, but not for networks with Poisson-like degree distribution. The strength of the paradox weakens only for the N_3 neighborhood, whereas for the N_1 and $N2$ neighborhoods is very strong and in a way competitive way among them. The observation that the paradox apprears stronger in N_2 for the simulated Barabasi-Albert network is not unrealistic since it is observed in the *CA-CondMat* network. 133

9.2 Evaluation of the influential spreading paradox at network level for the SIS and SIR spreading models in the *Email-Enron* network. The spreading paradox holding probablity is pretty high for the SIS model closely followed by SIR for the majority of the networks. The slightly lower paradox holding probablity for SIR is attributed to the existence of the R-state in that difussion model. Exceptions where the paradox does not hold are some very sparsely connected networks. The paradox holding probability is high in both N_1 and N_2 neighborhoods, which is a result observed for the centrality paradox as well. 136

9.3 Evaluation of the influential spreading paradox at individual node level for the SIS and SIR spreading models in the *Email-Enron* network. 137

9.4 Influence maximization under the SIR spreading model for the *Email-Enron* and *Brightkite* networks for the DEG, PCI and CORE centralities. 141

9.5 Influence maximization under the SIS spreading model for the *Email-Enron* and *Brightkite* networks for the DEG, PCI and CORE centralities. 142

9.6 Blocking the outspread of misinformation for the *Email-Enron* and *Brightkite* networks under the SIR propagation model for all centralities. 143

9.7 Evaluation of the centrality paradox at the individual level for the *Enron* network. Each line of plots corresponds to one centrality measure, namely DEG, PCI, CORE, PR and BC (the rest are given in the Supplement). Each column of plots corresponds to one neighborhood, namely the leftmost column is for 1-hop neighbors, the middle column of plots is about 2-hop neighbors, and the rightmost column of plots is about 3-hop neighbors. The x-axis in each plot depicts the size (in number of nodes) of the respective neighborhood, and the y-axis depicts centrality values. The heat values in the palette depict the centrality paradox holding probability. We observe that for a fixed neighborhood size, the centrality paradox holding probability decreases with increasing centrality value, for any centrality measure, and for all close neighborhoods. For some centralities, namely PR and BC this behavior is strictly 'binary', i.e., the centrality paradox either holds or not, no matter what the size of the neighborhood is. This binary behavior for all centralities is prevalent in N_2 and even more prevalent in N_3. 146

10.1 Overview of Map/Reduce and Hadoop (from [47]). 153

10.2 MapReduce pseudo-code for finding mutual friends. 155

10.3 MapReduce pseudo-code for finding connected components. 157

10.4 MapReduce pseudo-code for triangle counting. 158

10.5 Comparing TestDFSIO write throughput for 3 disks. 161

10.6 Comparing TestDFSIO read throughput for 3 disks. 161

10.7 (Left) CPU utilization for Connected Components with Orkut, using HDD, 1st iteration isolated. (Right) Disk usage for Connected Components algorithm with Orkut, using HDD, 1st iteration isolated. 166

10.8 (Left) CPU utilization for Connected Components with Orkut, using SSD2, 1st iteration isolated. (Right) Disk usage for Connected Components algorithm with Orkut, using SSD2, 1st iteration isolated. 167

A.1 Example of the k-shell decomposition method. 176

B.1 Out inter-degree distribution for the SLN networks when $d = 2$. 180

B.2 Out inter-degree distribution for the DLN networks when $d = 2$. 181

B.3 Distribution of in-out degree for the evaluated networks. Colored dots illustrate the percent of network nodes with the specific pair of (k_{in}, k_{out}) values. 182

B.4 k_{out} distribution of the layers for the semi-synthetic networks. 183

C.1 Paradox evaluation at the network level for all centralities, all neighborhoods and all networks. The y-axis illustrates the ratio $\langle c_{neigh} \rangle / \langle c \rangle$ normalized to all neighborhoods (N^1, N^2 and N^3). Negative values indicate that the network level paradox does not hold. It can be observed that moving from N^1 to N^2 favors the paradox, i.e., $\langle c_{neigh} \rangle / \langle c \rangle$ increases (i.e., strengthens the paradox) in most of the illustrated networks. Extending the evaluated neighborhood one more hop (to N^3) illustrates a decreasing trend (weakens). 185

C.2 Individual level centrality paradox for the *Brightkite* network in N_1, N_2 and N_3. . . . 186

C.3 Individual level centrality paradox for the *CA-Astroph* network in N_1, N_2 and N_3. . . 187

C.4 Individual level centrality paradox for the *CA-CondMat* network in N_1, N_2 and N_3 neighborhoods. 188

C.5 Individual level centrality paradox for the *CA-GrQc* network in N_1, N_2 and N_3 neighborhoods. 189

C.6 Individual level centrality paradox for the *CA-HepPh* network in N_1, N_2 and N_3 neighborhoods. 190

C.7 Individual level centrality paradox for the *CA-HepTh* network in N_1, N_2 and N_3 neighborhoods. 191

C.8 Individual level centrality paradox for the *Facebook* network in N_1, N_2 and N_3 neighborhoods. 192

C.9 Individual level centrality paradox for the *Hamsterster* network in N_1, N_2 and N_3 neighborhoods. 193

C.10 Individual level centrality paradox for the *PGP* network in N_1, N_2 and N_3 neighborhoods.194

C.11 Blocking the outspread of misinformation under the SIR spreading model for all networks. NP denotes the fraction of influenced nodes when there are no active blockers.195

C.12 Influence maximization for all networks under the SIR spreading model with cascade initiators biased towards the highest DEG nodes from N_1, N_2 and N_3 of RND. 196

C.13 Influence maximization for all networks under the SIR spreading model with cascade initiators biased towards the highest PCI nodes from N_1, N_2 and N_3 of RND. 197

C.14 Influence maximization for all networks under the SIR spreading model with cascade initiators biased towards the highest CORE nodes from N_1, N_2 and N_3 of RND. 198

C.15 Influence maximization for all networks under the SIR spreading model with cascade initiators biased towards the highest ONION nodes from N_1, N_2 and N_3 of RND. . . . 199

C.16 Influence maximization for all networks under the SIR spreading model with cascade initiators biased towards the highest CC nodes from N_1, N_2 and N_3 of RND. 200

C.17 Influence maximization for all networks under the SIR spreading model with cascade initiators biased towards the highest BC nodes from N_1, N_2 and N_3 of RND. 201

C.18 Influence maximization for all networks under the SIR spreading model with cascade initiators biased towards the highest PR nodes from N_1, N_2 and N_3 of RND. 202

C.19 Influence maximization for all networks under the SIS spreading model with cascade initiators biased towards the highest DEG nodes from N_1, N_2 and N_3 of RND. 203
C.20 Influence maximization for all networks under the SIS spreading model with cascade initiators biased towards the highest PCI nodes from N_1, N_2 and N_3 of RND. 204
C.21 Influence maximization for all networks under the SIS spreading model with cascade initiators biased towards the highest CORE nodes from N_1, N_2 and N_3 of RND. 205
C.22 Influence maximization for all networks under the SIS spreading model with cascade initiators biased towards the highest ONION nodes from N_1, N_2 and N_3 of RND. . . . 206
C.23 Influence maximization for all networks under the SIS spreading model with cascade initiators biased towards the highest CC nodes from N_1, N_2 and N_3 of RND. 207
C.24 Influence maximization for all networks under the SIS spreading model with cascade initiators biased towards the highest BC nodes from N_1, N_2 and N_3 of RND. 208
C.25 Influence maximization for all networks under the SIS spreading model with cascade initiators biased towards the highest PR nodes from N_1, N_2 and N_3 of RND. 209
C.26 Evaluation of the spreading paradox at the individual level for the SIR spreading model for the following networks: Brightkite, CA-AstroPh, CA-CondMat, CA-GrQc, CA-HepPh. 210
C.27 Evaluation of the spreading paradox at the individual level for the SIR spreading model for the following networks: CA-HepTh, Facebook, Hamsterster, PGP. 211
C.28 Evaluation of the spreading paradox at the individual level for the SIS spreading model for the following networks: Brightkite, CA-AstroPh, CA-CondMat, CA-GrQc, CA-HepPh. 212
C.29 Evaluation of the spreading paradox at the individual level for the SIS spreading for the following networks: CA-HepTh, Facebook, Hamsterster, PGP. 213

Part I

Introduction

CHAPTER 1

INTRODUCTION

1.1 Motivation

The scientific study of complex networks, i.e., social networks, biological networks, computer networks, technological networks etc., combines ideas, tools and methodologies from a wide range of different research areas that includes computer science, mathematics, social sciences, biology, physics and more [144]. It is an interdisciplinary area of science that sheds light to networked environments by cross-sharing knowledge and conclusions. The work of the current thesis presents an extensive study in such networked environments that includes single networks [77], multilayer networks [61] and networks of vehicles [154]. All these systems will be thoroughly explained throughout this document from the perspective of network science. A substantial portion of this dissertation leverages tools from graph theory and epidemiology to study dynamical processes taking place over various real and simulated networks such as the spread of information, the outspread of malicious data and the identification of network nodes that play a crucial role in such processes.

Specifically the present thesis poses and answers−among others−several questions and challenges: *how can we detect nodes-people-vehicles that can spread information rapidly within a network; Can we detect such potent entities based solely on local knowledge of the network topology; and thus effectively deal with rapidly changing networks or incomplete knowledge of a network's connections; Are those measures both effective, i.e., easily applied, and efficient; How such local approaches fare against metrics that consider the entire network topology; Can these measures be redefined to address those questions in more complex structures such as the multilayer networks; and if so how do they fare; How can the friendship paradox be utilized as a sampling methodology to detect a group of nodes for maximizing the outspread of information in modern* *social networks with minimum computational cost; Alternatively how can we distinguish nodes capable of stopping or hindering the outspread of undesired data within such networks;* Prior to answering the aforementioned issues we must first take a glimpse in several separate components that constitute the guiding force of this research, i.e., the networks and network types, employed tools from graph theory and the spreading models.

Networks

A single network, in its most generic form, is composed of a set of nodes (vertices) connected physically or logically in pairs by a set of links (edges). Edges can be undirected or directed−in the sense that a node points to another and the node may be or may not be pointed back−and may be associated with a weight, that could represent the cost of transmission between nodes or the amount of data flowing between them, etc. With this definition an outstanding amount of real life examples emerge; the Internet, Social networks (Facebook, Twitter, etc.), Transportation networks, Power Grids, Citation networks, Collaboration networks, Neural networks, Biological networks and so on. Hence it is self explanatory why the research communities from diverse disciplines dedicated so much effort in the study of such connected structures, i.e., the nodes, the edges and the pattern of those connections. Evidently, if we are able to gather the necessary data for the representation of a networked structure, inherent questions arise: how can we use those data; what conclusions can be made for the system the network represents; and how can we exploit any emergent network properties related to the practical issues that concern our particular system?

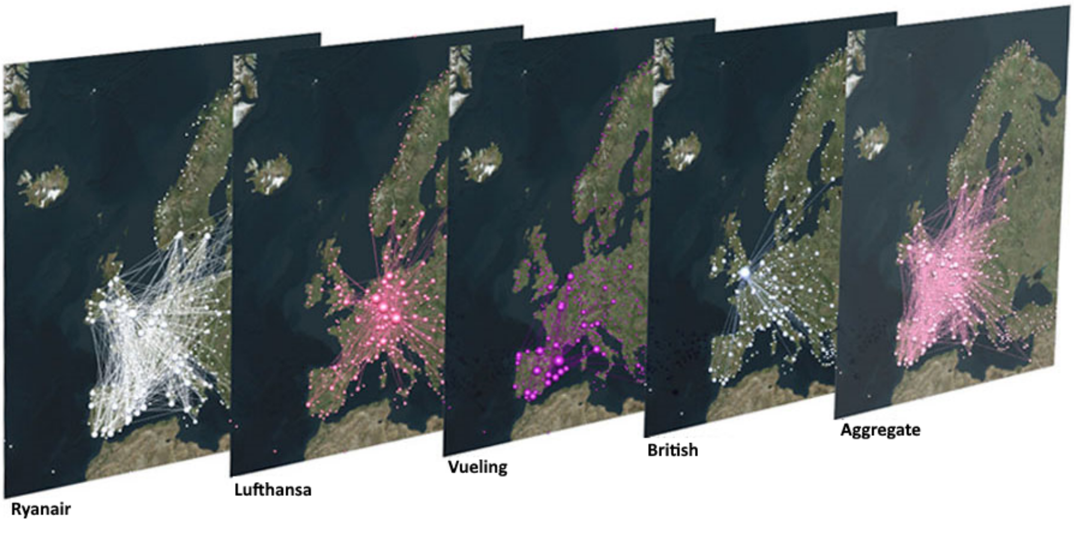


Figure 1.1: Multiplex network of European airlines.

Single networks however are an oversimplification of more complex systems known as *multilayer interconnected networks* [61]. In these structures−unlike single networks−nodes may be connected with multiple type of connections−each edge type corresponding to a different *layer*–or may be associated with nodes that belong to entirely different networks (different layers) and thus form what is known as *networks of networks*. For example the airlines transportation network is a multilayer network (Figure 1.1) where any two cities (nodes) may be connected with multiple airline agencies (different type of connections). In this case a network layer is represented by the edges (airline schedules) of a single ariline agency (e.g., British airways), and all such layers form the multilayer network. Another example is that of social networks where a user-node may have accounts in different social platforms (layers) such as Twitter, Facebook, LinkedIn, etc., and thus be connected to all these networks. The common characteristic of the aforementioned examples is that the nodes are counterparts of themselves in the different layers, i.e., the same city or the same user respectively. These networks where nodes are “clones” in the different layers are also known by another name, that is, *multiplex networks* [73]. The family of multilayer networks encompasses a variety of different layered structured networks some of which are presented in this thesis. These complicated systems are known to the research society for decades but the massive generalization of the large body of knowledge from graph theory to multilayer networks is a recent phenomenon.

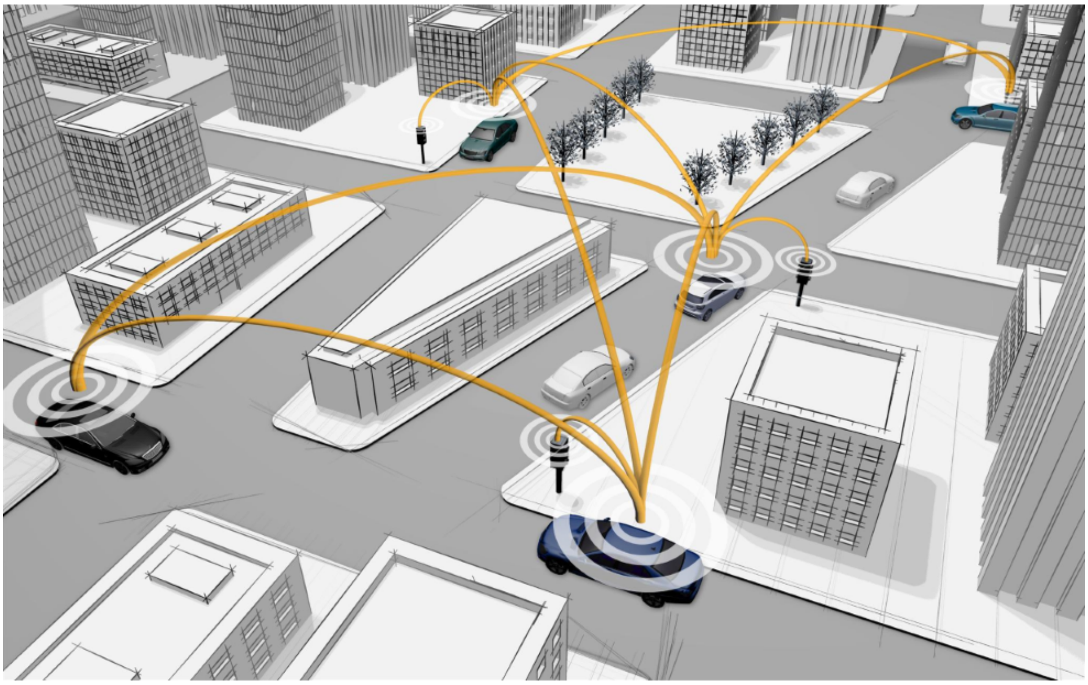


Figure 1.2: A network of vehicles and road side units.

Finally, thanks to the advances in wireless technologies and the automotive industry, vehicular ad-hoc networks (VANETs) have become a promising research field. Simply put, it is a group of vehicles equipped with wireless interfaces, able to communicate with each other and nearby stationary equipment usually referred to as road side units (RSUs) as depicted in Figure 1.2. The introduction in the automotive market of vehicles with wireless communication capabilities that will allow a vehicle to communicate with other vehicles in vicinity (vehicle-to-vehicle communication, V2V) will bring a revolution in sectors such as vehicle/driver safety [57], Internet access and entertainment [134]. V2V systems are particularly appealing for the vision of the “always connected car", because a fully functional V2V system would connect drivers traveling near each other, allowing a vehicle to accumulate information about what other vehicles are doing even if the driver can not see them. The prospects of this technology is truly tremendous, from practically eliminating human casualties, to reducing traffic congestion, or setting up vehicular computing clouds to exploit the aggregate computing and storage capability of roaming cars. NHTSA estimates that this technology can prevent up to $592,000$ crashes and save $1,083$ lives per year[1]. The Crash Avoidance Metrics Partnership (CAMP) is already working on creating common standards and a common technology for automakers to use so as they release fully functionall vehicles with V2V capability in the next years. The vehicular environment poses significant challenges to the research community mostly due to the dynamic mobility of vehicles and potential obstacles−such as buildings−interfering with the communication. Nonetheless, the vehicular network has attracted huge research attention due to the numerous benefits brought to the society by the cumulative knowledge build by such a network.

Many aspects of those networked structures are worthy to study. For instance the pattern of connections between computer nodes in the Internet depict the routes that data take while traversing the network, whereas connections in social networks illustrate how networked populations learn from the opinions of their peers, gather news, as well as phenomena such as the spread of a disease. Nowadays modern networks have grown enormously large−hundreds of thousands or even millions of nodes, vertices, actors−and the vast scientific background of network science has developed an arsenal of measures, metrics and techniques that can help us study and understand what our network data portray and their properties. The present thesis has contributed to this body of knowledge by defining and evaluating new measures for single networks as well as their redefinition for multilayer and vehicular systems.

Centrality Measures

An example of an important and useful class of network measures that has also been part of the core of this dissertation is that of *measures of centrality*. Centrality quantifies the importance of nodes or edges in a networked system and there is a wide range of such measures that captures different aspects−different interpretations of importance−of a node (or edge). The most straightforward but very useful metric is the *degree* centrality which quantifies the number of incident edges upon a node. Although simple, it has found its use in many applications. Of particular importance are the *hub* nodes−nodes with unusually high degree−that play a crucial role in the functioning of the network, e.g., robustness, when network nodes fail or malfunction,

[1] http://www.nhtsa.gov/About+NHTSA/Press+Releases/2014/NHTSA-issues-advanced-notice-of-proposed-rulemaking-on-V2V-communications

or for the efficient diffusion of information within the network. The *geodesic distance* for a pair of nodes−the minimum number of edges in-between two vertices−is also a very popular measure. Evidently measuring the shortest distances between pairs of vertices is of paramount importance and has given birth to various measures such as *betweenness* and *closeness* centrality. Betweenness centrality [93] captures the number of shortest paths that traverse a certain node or in other words, how much a particular vertex falls between others, whereas closeness centrality [195] quantifies the mean shortest distance from a node to all other nodes. Both approaches have been used in applications−among others−regarding the spread of information in social networks. For instance nodes with large closeness index might have better access to information from other nodes or exert more direct influence towards them, whereas nodes of high betweenness may have considerable control over the information passing between others.

PageRank [171] is another widely used centrality measure named by the Google web search corporation and used as a core part for ranking the web-pages of the World Wide Web. Briefly an important web page is one "pointed" by many important pages which by virtue will have a large PageRank index. In the literature it has found fertile ground for use in many applications such as detecting the most important scientific papers in citation networks and has been a benchmark approach for the devise of other metrics such as TwitterRank for detecting important-influential Twitter user-accounts. The *k-core* decomposition [139] of a network is yet another very popular algorithm that categorizes nodes in cores (shells). In short, it is a pruning mechanism that removes nodes from the network that are not at least k connected, meaning that all nodes in the k^{th} shell have at least k degree. It has been widely used for the identification of influential spreaders in complex networks, has given rise to many other techniques addressing its disadvantages [18], [105], [122] and has been one of the major competing algorithms for a set of proposed centrality measures presented in this dissertation.

The *μ-Power Community Index* (μ-PCI) [77] has been a subject of intensive study in the present thesis and is a centrality measure that characterizes a node for the density in connections of both itself and its μ-hop neighbors. In the upcoming chapters the reader will become more familiar with μ-PCI, its applications in complex networks [150] and its spreading dynamics. It is an effective−low computation cost−and efficient method for detecting potent nodes in large complex networked systems that can spread data (information, advertisements, rumors, etc.) to a large subset of network nodes. The research community has devoted much effort in redefining centrality measures to fit the domain of multilayer networks: the multiplex PageRank [87], versatile PageRank [29], multiplex Betweenness [29] and the k-core percolation [48]. Part of this dissertation also studies the behavior of those redefined metrics, proposes a family of novel approaches based on μ-PCI for multilayer networks and evaluates their performance for accelerating spreading processes in both real and generated networked systems.

The centrality measures presented so far is only the tip of the iceberg of an almost unlimited literature of such metrics and their utilization. The interested reader is referred to [73][29][38] and references therein. Among their many applications in real systems, their role in dynamical processes such as the spreading dynamics has attracted excessive attention from the research community. Their role as information spreaders (or inhibitors) is at the core of this dissertation where nodes with large centrality index (e.g., hub nodes) may act as cascade initiators-originators to accelerate (decelerate) the propagation of information in real complex networks.

Spreading Dynamics

Understanding the spreading dynamics in complex networks to either boost the spreading of information or stop the outspread of undesired "things" is a core part of study for this dissertation. In the literature, spreading phenomena are correlated with the outspread of infectious diseases from epidemiology and at its core lie the epidemic models. Epidemic modeling is thus an interdisciplinary research area that has developed a wide variety of approaches ranging from simple explanatory models to very elaborate stochastic methods and rigorous results [106]. Here, the epidemic modeling metaphor has been introduced to describe a wide array of different phenomena in real networks such as the spread of information, cultural norms, rumors, social behavior, malware etc., all such phenomena modeled as a contagion process whose mathematical description relies on classic epidemic models.

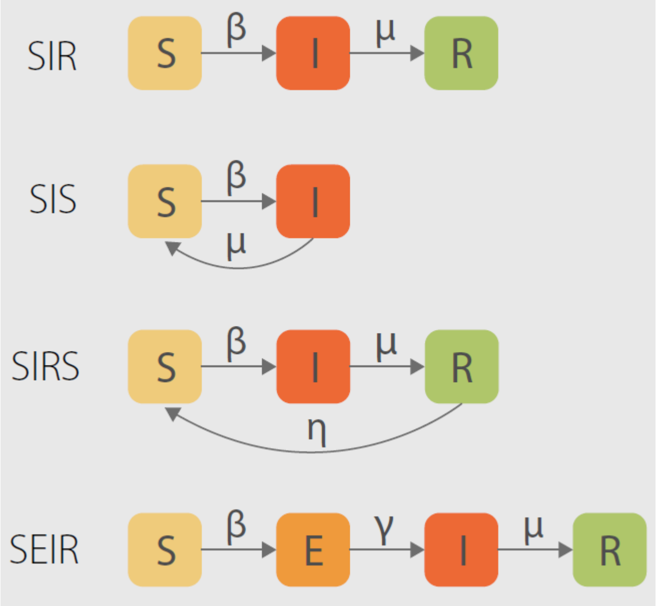


Figure 1.3: Transition probabilities between the different states of various spreading models.

The current thesis is focused in only a few models (the most widely used ones), namely the *Susceptible-Infectious-Recovered* (SIR) and the *Susceptible-Infectious-Susceptible* (SIS) models, that have also been the core for the development of other newer spreading models. The former includes three distinct states; the infectious state (I) where a node tries to infect (influence) some (or all) of its direct neighbors, the susceptible state (S) where a node can be infected (influenced) by an I node and the recovered or immune state (R) where nodes cannot be infected (influenced). For SIS however there are no recovered nodes and hence a node can be either infected or susceptible. In both models the transition between the different states is associated with some given probability often related to network properties such as the largest eigenvalue of the network's adjacency matrix [106]. SIR (like mumps) instructs permanent immunization to the network nodes and measures how far a virus (meme) will "travel" in a network, i.e., how many nodes will become influenced, whereas SIS (like flu) quantifies its ability to "preserve" itself within a network, i.e., whether a meme will manage to keep "interested" a substantial portion of network nodes or it will eventually loose interest and die out. Other popular models are the SIRS model where nodes are not permanently immune and hence become susceptible after a period of time or the *Susceptible-Exposed-Infectious-Recovered* (SEIR) model where a node after infection remains in the exposed (pre-infected) state before entering the I state to start its contagious behavior. Figure 1.3 depicts the transition states of the spreading models. For more details please refer to [106].

Hence, given a network of contacts−whom-links-to-whom−and a set of initially incentivised (infected) nodes acting as originators of a diffusion process, *will a contagious product/meme/virus spread and become epidemic, i.e., affect a large subset of network nodes, or will it die-out quickly; What will change if nodes have partial, temporary or permanent immunity; Which are those topological characteristics that a network node should have to cause an epidemic spreading; Can centrality metrics be our guide for detecting such influential nodes;* These generic questions bind together several aspects of research conducted in this thesis and will be discussed in detail in the upcoming chapters.

1.2 Synopsis

In this thesis, we take a trip in network science and graph theory to better understand the behavior of the systems networks represent. We focus on the structure of networks, and study dynamical processes taking place on networked environments that bear unique characteristics, that is, probabilistic, single, multilayer and vehicular networks. A wide range of tools from graph theory is employed and accompanies the different phases of this dissertation. In Part II we study how to effectively and efficiently select influential nodes in networked systems, and evaluate a series of proposed methodologies in real complex networks, for their ability to accelerate the spreading process. Part III deals with the opposite problem where our objective is to hinder the outspread of undesired data in complex systems. In Part IV, based on conclusions drawn from social sciences we propose a strategy for sampling network nodes from networks of colossal volumes of data (e.g. Twitter, Facebook, LinkedIn, etc.) for the design of better influential node detection algorithms. Finally in Part V we compare the performance of solid state and hard disk drives for social network analysis in the Hadoop ecosystem.

Overall the present thesis has contributed to the body of knowledge in network science in several domains. Among the various chapters that will be introduced, we make a quick note of our most significant contributions:

- we proposed a centrality metric for detecting efficient spreaders in complex networks, based solely on local knowledge of the network topology, and thus suitable for rapidly changing networks, large scale networks, temporal networks and real time applications. The proposed metric was evaluated in diverse real network typologies and was rendered superior in identifying more efficient spreaders compared to the state-of-the-art methodologies. Related publication [77].
- we generalized the well established h-index [23] algorithm in the domain of multilayer and multiplex networks. Specifically, a family of centrality metrics have been introduced, with the h-index methodology imprinted within the core of the proposed methods, each bearing its own limitation and advantages in the multilayer structure. The proposed techniques, were evaluated for the identification of influential spreaders, i.e., nodes that can spread information to as many layers as possible and as many nodes as possible within each layer, in a wealth of real and synthetic multilayer networks. Related publication [10].
- inspired by the intuition brought by the friendship paradox–*your friends have more friends that you do*–we have generalized and empirically shown that the concept of neighbor superiority [89] also applies for a wide range of centrality metrics, that is, your friends are more central than you. Furthermore we provide solid proof over a wide range of simulation results in real complex networks, that the paradox sense, also applies for probabilistic characteristics such as the ability of nodes to spread information over a network, i.e., *your friends can spread information more efficiently than you*. The findings of this study can straightforwardly be applied for mining efficient spreaders in gigantic real networks and in designing better influential node detection algorithms. Related work [S1].

Chapters Overview

In **Chapter 2** we propose a centrality metric–an amalgam of node degree and betweenness–that based solely on local knowledge of the network topology, can efficiently rank nodes with respect to their spreading capability. Its local computation cost, renders the proposed technique appropriate for real time application and dynamically/rapidly changing networks. The evaluation conducted in real complex networks, illustrates the superiority of the proposed technique for identifying more accurately influential spreaders, compared to the state-of-the-art solutions.

Chapter 3 embraces the probabilistic nature of several real world networks where connections are not static but rather dynamic. The probabilistic characteristic implies that any two connected nodes share a common "time span" where information can flow between them, e.g., in online social networks, and hence any such link is active only on that duration. Given a real network, the probabilistic attribute is tuned per network link by following a Zipfian distribution, providing us the capability to generate various network links, ranging from very weak ties to very strong ones. Here we introduce a methodology which measures the number and strength of limited length paths (e.g., 2, 3 and 4 hops) that emanate from each node, and synthesize a measure of probabilistic importance. The corresponding framework illustrates the dynamicity of the proposed technique to handle the variability of probabilistic edges and effectively detect influential spreaders in probabilistic complex networks.

Chapter 4 analyzes information diffusion processes in multilayer interconnected networks and proposes a family of centrality measures able to rank a multilayer (or multiplex) node, with respect to its spreading capability in these systems, i.e., its ability to spread information to as many layers as possible and to as many nodes as possible in each layer. The proposed technique is a generalization of the well known h-index [23] centrality to multilayer networks and each proposed method poses it's unique advantages and limitations. The evaluation conducted illustrates the ability of the proposed methods to detect efficient spreaders in multilayer environments by outperforming a wide range of competing techniques in real and semi-synthetic networks.

Chapter 5 focuses on the vehicular environment where we search for efficient relay vehicles that can maximize the outspread of information in a network of vehicles. Each vehicle-node collects information for its surrounding vehicles with the exchange of regular *heartbeat* messages. For each vehicle in range (each neighbor) a vehicle holds information regarding its direction, its speed, the quality of link between them as well as its neighbor list. A centrality metric that utilizes the heartbeats information is proposed to create limited length walks (similar to Chapter 3) that emanate from each vehicle and define its importance in the network. The simulation results in diverse vehicular conditions show that the proposed algorithm can efficiently diffuse information in a network of vehicles.

Chapter 6 addresses the outspread of malicious information (e.g., a meme, virus, etc.) in networked populations with aim to hinder its propagation over the network. We shift the problem to the level of edges, whose removal can mitigate the diffusion of malicious data to the largest possible extent. We follow the spreading process as it progresses in time, and create limited length subgraphs, emanating from each newly infected node. In the resultant subgraphs, the dynamicity of each edge is measured by quantifying the number of shortest paths that traverse each particular edge within each subgraph. The proposed methodology provides the overall ranking for those edges that lead to susceptible nodes and are potential candidates for removal in each spreading step. Our evaluation showed that the proposed mechanism outperforms a set of competitors in a wealth of real world networks, by more effectively hindering the malicious propagation, i.e., by protecting more network nodes.

Chapter 7 investigates the propagation of software viruses over a vehicular network. We propose a distributed solution to block the outspread of the virus within a network of vehicles, by initiating a negating spreading process−triggered by a centralized unit and spread across the network by vehicles−that informs vehicle-nodes for the presence of infected ones. Each vehicle holds a list of all−so far identified−infected sources as well as a list of potentially infected vehicles, which are circulated between opportunistic neighbors encountered in the vehicular environment and block communications between them. The simulations conducted employs a realistic environment from the city of Erlangen in Germany, with a rich road topology of regular intersections, and buildings that interfere with the communication. The results illustrate that the proposed mechanism can efficiently mitigate the outspread of a virus in a vehicular network.

Chapter 8 discusses the performance of vehicular applications in the presence of false data. Specifically the proposed work is a combination of V2V, V2I, and I2I communications for rerouting vehicle nodes based on their desired destination, CO_2 emissions and potentially congested road segments along the way. We employ several scenarios, where we inject false data in the proposed system regarding the road conditions that these malicious vehicles experience, and propose a defense mechanism based on the cumulative knowledge built from the collaborative vehicles, to filter out such spurious data. The extensive simulations conducted over various road conditions and different methods of injecting false data, shows that the proposed mechanism can distinguish spurious data and restore the performance of the system to its normal behavior.

The research in **Chapter 9** employs the friendship paradox (*your friends have more friends than you*) into the domain of centrality measures and power of influence. In this study we empirically show that the paradox intuition applies also for a wide range of centrality metrics, i.e., your immediate connections have higher centrality than you, as well as in the influence domain, i.e., your friends are more influential than you. The findings of this investigation can straightforwardly be used for designing better influential nodes detection algorithms, e.g., by refraining from selecting as initial spreading nodes those who are neighbors or for estimating the spreading capability of nodes using their friends' capability. Our evaluation over a wide range of real networks supports our claim for employing the paradox intuition (centrality/influence paradox) for accelerating/decelerating the spreading process.

Chapter 10 performs an empirical evaluation on the performance of solid state drives (SSDs) against the performance of hard disc drives (HDDs) in the Hadoop environment for the analysis of social networks. Specifically, the Hadoop platform is used for the processing of big data to run computationally intensive analytics such as finding mutual friends, counting triangles and calculating connected components in colossal volumes of network data (e.g., Facebook, Youtube). The conclusions drawn from our evaluation indicate that blindly adding SSDs to Hadoop is not an appropriate solution, but rather build components for assessing the type of processing pattern of the application and then direct the data to the appropriate storage medium.

Part II

Spreading Dynamics in Complex, Multilayer and Vehicular Networks

CHAPTER 2

ACCELERATING SPREADING PROCESSES IN SINGLE COMPLEX NETWORKS

Detecting Influential Spreaders in Complex, Dynamic Networks

2.1 Introduction

In this chapter we focus on the problem of influential spreaders—nodes in complex networks that can spread a message rapidly among other nodes. Early detection of such entities can help security technologists prevent extended damage to networks against malware or, in the case of terrorist networks, identify the most important malefactors. To identify influential spreaders, researchers traditionally have relied on the k-shell index [139], a degree- based measure of a node's "coreness." However, the significant computational overhead of this index makes it inappropriate for analyzing dynamic networks. We propose an alternative measure, the μ-power community index, that is an amalgam of coreness and betweenness centrality; μ-PCI is calculated in a completely localized manner and thus suitable for any kind of network irrespective of its size or dynamicity [138]. An experimental evaluation of the two values, along with a baseline measure based solely on node degree, demonstrates μ-PCI's superiority in detecting influential spreaders.

Related publication [**J3**]: Pavlos Basaras, Dimitrios Katsaros, Leandros Tassiulas. *Detecting Influential Spreaders in Complex, Dynamic Networks*, **IEEE Computer magazine**, vol. 46, no. 4, pp. 26-31, April, 2013.

2.1.1 Motivation

Consider an example in which an attacker installs a virus on a host mobile device, with the intention of exploiting the host's connections to spread the malware, and ultimately infect as many other devices as possible. Assume that all devices comprise a single network with common administration. Upon detecting the malware, the administrator immediately takes action to limit its propagation. Possible measures include installing more effective antivirus software to selected devices, shutting these devices down, or disconnecting them from the rest of the network.

Two well-known cases of malware that exploit mobile devices' network connections are the Cabir and Commwarrior-A worms. The former spreads through Bluetooth connections to other Bluetooth-enabled devices that it can find. The latter was the first worm to propagate via the Multimedia Messaging Service; it searches through a user's local address book for phone numbers and sends MMS messages containing infected files to other users.

Obviously, if the infected devices in our scenario are influential spreaders, they will impact a large part of the network. This leads to several questions: How fast will the virus spread? Is the infection rate different in different network topologies? Does the percentage of infected nodes in the network depend on the node(s) where the infection originated? Do multiple infection starting points produce a substantially broader infection area? If so, what does this depend on? Which nodes should the administrator disconnect to stop the propagation?

Researchers who have investigated such questions found that not all nodes in a complex network have the same potential to propagate a message efficiently [111], [139]. Explanations for this behavior range from a network's topological characteristics at global scale—for example, power-law degree connectivity—to individual nodes' connectivity patterns.

2.2 Identifying Influential Spreaders

Most studies of influential spreaders have focused on their linkage with other nodes. The problem has not been described formally but is similar to two others: detecting a network's central nodes and selecting the set of nodes that maximize the spread of infection.

Identifying the central nodes in a complex network usually relies on graph-theoretic concepts of betweenness centrality. Such measures are generally based on a node's degree or on its geodesic distance to other nodes [194]. The former category includes degree centrality, spectral centrality, and coreness, whereas the latter includes closeness, shortest-path, and bridging centrality. Degree-based centrality measures consider a node prominent if its connections make it visible to the network's other nodes. Intuitively, a node is prominent if it is adjacent to many other prominent nodes. The latter family of centrality measures exploits the shortest path between nodes.

The spread maximization problem has been proved to be NP-hard in threshold networks [183], and researchers have proposed several greedy algorithms to solve it—for example, there are simple and efficient algorithms that adopt the voter model.

Recent studies of social networks have considered other node features besides connectivity such as age, gender, and marital status [107]. Another feature is trustworthiness, which can affect a decision to follow a link to malware. Examples of malware that exploited trust to spread across a social network include the Skype and Koobface worms.

2.3 Balancing Betweenness and Coreness

Maksim Kitsak and his colleagues found that the degree of a node is not a good indicator of its ability to spread a message to a sufficiently large part of the network [139]. Furthermore their work showed that measures based on betweenness centrality are distorted by the degree-1 node, which increases the centrality index of the sole node connected to them. Our own research found that exploiting betweenness centrality has several disadvantages for disseminating messages in wireless ad hoc networks [138]. Relying on a degree-1 node results in overestimating the spreading capabilities of a node connected to it. Moreover, based on a detailed investigation of the spreading capabilities of high-degree nodes in various complex networks, we found that high-degree nodes are indeed often good spreaders.

Kitsak's team argued that the node's position in a k-shell decomposition (see Appendix A.2) of the network's graph is a better way of quantifying influential spreaders, and went on to verify this hypothesis in the context of disease propagation [139]. However, subsequent research proved that a node's spreading capabilities in the context of rumor spreading do not depend on its k-core index [109]. The "K-Shell Decomposition" of a network has two other major shortcomings. First, it has significant computational overhead, rendering it unsuitable for dynamic networks. Second, it is impossible to guarantee a monotonic relationship between the k-shell index and a node's spreading capability, which causes major problems when there are not enough resources to expend on node vaccination.

We have developed a method that quantifies spreading capabilities in a completely localized manner, making it suitable for any kind of network irrespective of size or dynamicity [138]. This metric, μ-PCI, balances the principles of betweenness centrality—it considers nodes that lie on many communicating paths between pairs of nodes—and the transitive network density implied by the coreness measure. The metric is computed as follows: *the μ-PCI of a node v is equal to k, such that there are up to $\mu \times k$ nodes in the μ-hop neighborhood of v with degree greater than or equal to k, and the rest of the nodes in that neighborhood have a degree less than or equal to k*. The goal is to detect nodes located in dense areas of the network and thus likely influential spreaders.

2.4 Performance Evaluation

To evaluate our technique's accuracy, we compared it to the k-shell decomposition and a baseline measure based solely on the node degree, over several complex networks. Here, we present the most significant findings from two well-known networks, CA-CondMat and CA-AstroPh—collaboration networks from the e-print arXiv, covering condensed matter physics and astrophysics, respectively—from the Stanford Network Analysis Platform [65]. Table 2.1 summarizes the networks' main characteristics.

Table 2.1: Complex Network Attributes

Network	**Type**	**No. of Nodes**	**No. of Links**	**Infection Probability (%)**
CA-CondMat	Sparse	23133	186936	8
CA-Astroph	Dense	18772	396160	4

We used the Susceptible-Infected-Recovered (see Appendix A.1) model (SIR) to investigate the spreading process. For the spreading probability, namely λ, we employ relatively small values to highlight the importance of influential spreaders. Our performance evaluation considered infections originating with both a single spreader and multiple spreaders.

Sinlge Original Spreader

All nodes are initially at the susceptible (S) state, except for one node which is in the infected (I) state. The infected node tries to infect its susceptible neighbors with probability of success λ, and immediately after enters the recovered (R) state. All nodes in state I try to infect their susceptible neighbors, and the process repeats until there is no node in the I state.

Multiple Original Spreader

The number of initially infected nodes ranges from 0.5 to 4 percent of the network's total size. μ-PCI and node degree methods share a similar selection procedure. The malicious set of spreaders is empty in the first phase. We introduce the spreader with the highest value of each method to its respective set, and then select the spreader with the next highest value, which is not directly connected to the previous set. The process repeats until the initial infection percentage of the network is satisfied. For the k-shell decomposition, all spreaders in each shell are treated evenly, hence, we start by selecting a random node among the nodes residing in the highest k-shell. We then randomly select the next spreader from the remaining nodes of the core shell, that are not directly connected to the previous set, and continue this process iteratively. If the initial infection percentage cannot be met from the core shell, we repeat the process on the shell immediately below it, and so on.

For μ-PCI, we present only results for $\mu = 1$. We obtained analogous results for $\mu = 2$, but the method's performance deteriorates substantially for $\mu > 2$. We use km, ks, and k to represent the 1-PCI, k-shell index, and node degree values, respectively. Similar to Kitsak and his colleagues [139], we used the average size of the network's infected area as a performance measure. To quantify inf(s), the influence of a single spreader s, we computed the average size of the network infected with the (km, k) pair values. We averaged the extent of the infected network over all spreaders as follows:

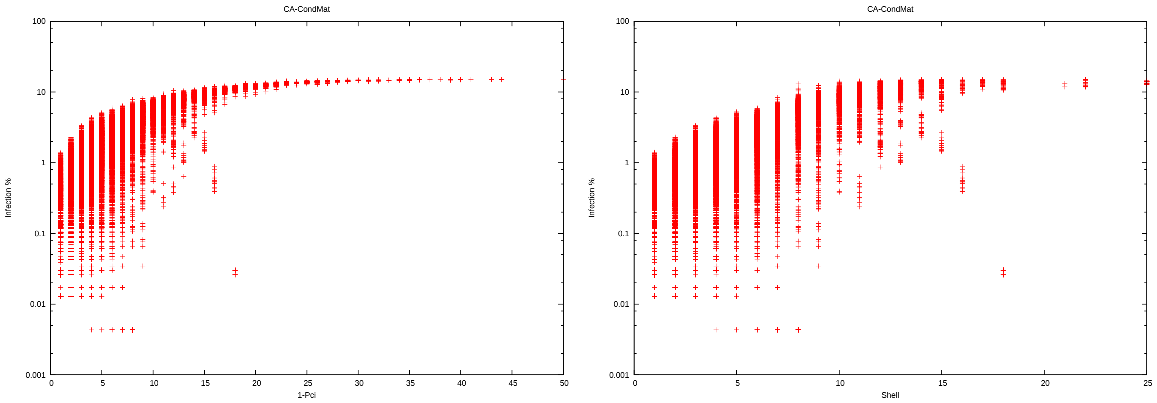


Figure 2.1: Spreading capability of nodes in the ca-CondMat network with a single original spreader according to (a) 1-PCI and (b) k-shell index. There are nodes with high k-shell indices, some of which infect a large portion of the network, as well as nodes with the same k-shell index (16) that infect a significantly smaller part of the network. On the other hand, only nodes with very small 1-PCI exhibit such behavior.

$$INF_{km,k} = \sum \frac{inf(s)}{N_{km,k}} \tag{2.1}$$

where $P_{km,k}$ is the set of all $N_{km,k}$ spreaders with the same (km, k). We repeated the same process for k-shell decomposition.

To obtain statistically unbiased results, we repeated the computation 1,000 times for each vertex of a graph for the single- and multiple-origin scenarios. We found that 1-PCI exhibits steady and reliable behavior, overcoming the disadvantages of high- degree spreaders and of k-shell decomposition. Choosing high 1-PCI nodes maximizes spreading influence, whereas selecting the high-degree nodes or a random node from the core shell either results in poor spreading or does not maximize influence.

2.4.1 Single original spreader

Our first experiment examined the three methods' ability to select the most influential spreaders for a single-origin process.

Figure 2.1 shows all nodes' spreading capability in the CA-CondMat network according to their 1-PCIs and k-shell indices. The 1-PCI method results in a more monotonic distribution than k-shell decomposition, providing a clearer ranking of spreading capabilities. It converges to an approximately straight line, where maximum influence lies, more steeply than the k-shell method in all studied cases. Choosing a spreader with, say, 1-PCI > 23, will yield the maximum influence, whereas choosing one from the core or from the high shells might not be optimal, because in some cases nodes within the same shell have different spreading capabilities.

Figure 2.2 shows the spreading capability of all nodes in the CA-AstroPh network according to their 1-PCIs and k-shell indices, versus the respective node's degree. In particular, the plots

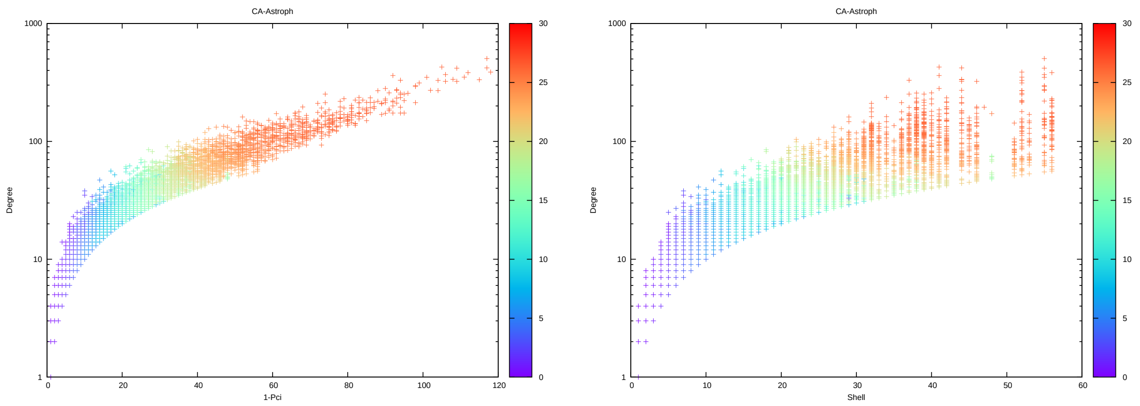


Figure 2.2: Spreading capability of nodes in the CA-AstroPh network with a single original spreader according to (a) 1-PCI and (b) k-shell index versus node degree. The k-shell index fails to fulfill monotonicity in many cases, and 1-PCI has a better correlation with node degree.

depict the average size of the infected population $INF_{km,k}$ for all spreaders with (1-PCI,degree) pair values. The k-shell index clearly fails to fulfill monotonicity in many cases. Also, 1-PCI has a better correlation with node degree.

This experiment confirmed the conclusion of Kitsak's team that measures such as node degree cannot accurately predict a network's most influential spreaders [139]. For a fixed degree equal to k, there is a wide spectrum of $INF_{km,k}$ values, making the degree measure an ineffective solution, especially in cases where the objective is to select a very small number of spreaders. This occurs because a high-degree node might be located in a sparse neighborhood. The k-shell index depends less on node degree when moving to higher shells, but the best spreaders are often scattered across numerous shells, thus violating monotonicity. For instance in Figure 2.2(right) we observe that nodes with a k-shell index equal to 48−which is particularly high−have spreading capability similar to that of nodes with a k-shell value less than 30. For a fixed 1-PCI, the infection percentage is approximately the same and independent of node degree, making high 1-PCI nodes the best choice in single-origin spreading processes. The 1-PCI measure groups spreaders according to their spreading capabilities: lower 1-PCI values correspond to poor spreaders, whereas high values indicate the most influential ones. As a node's 1-PCI increases, its spreading influence also appears to increase. Consider, for example, the results obtained from the CA-AstroPh network shown in Figure 2.2(right). Moving to higher shells—starting at, say $ks > 34$—spreading influence seems to constantly increase. However, this increase stops at $ks = 48$, where the infection decreases drastically. The 1-PCI analysis does not elicit such behavior, especially when close to maximum influence. As 1-PCI values increase, influence also continuously increases until maximum infection is reached.

We computed the number of influential spreaders that can achieve the maximum infection (with 1 percent deviation) for the two networks described here along with the soc-Slashdoc0811 network. As Table 2.2 shows, network size and topology impact the number of influential spread-

Table 2.2: Number of influential spreaders that can maximize infection in three networks.

Network	Size (nodes)	Densisty (edges)	Infected area (%)	No. of influential spreaders
soc-Slashdoc0811	77360	905468	16.5	1788
CA-CondMat	23133	186936	1.9	127
CA-Astroph	18772	396160	26.5	477

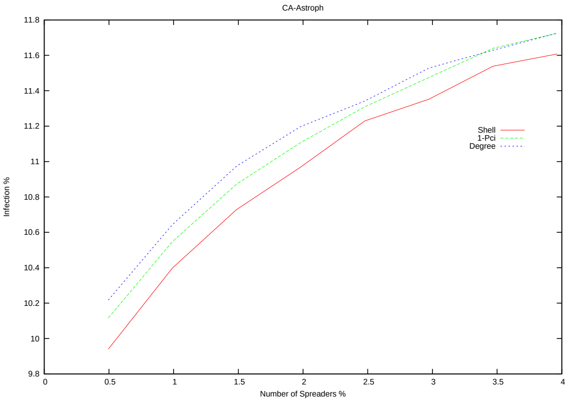


Figure 2.3: Spreading capability of nodes in the ca-AstroPh network with multiple original spreaders according to node degree, 1-PCI, and k-shell index. The k-shell index is the least effective measure. Node degree is the most effective measure, closely followed by 1-PCI, but the discrepancy between these values quickly diminishes as the number of multiple original spreaders grows.

ers. We observed no increasing or decreasing relation between the number of influential spreaders and network size—the key factor is the pattern of node connections.

2.4.2 Multiple original spreader

Our second experiment examined the three methods' ability to select the most influential spreaders for a multiple-origin process. To maximize the infected area, the original spreaders were not linked. If the selected spreaders were connected, the infected region would be smaller due to the overlap of neighboring spreaders' "influence regions" [139].

Figure 2.3 shows all nodes' spreading capability in the CA-AstroPh network according to their degree, 1- PCI, and k-shell index. The x-axis indicates the percentage of initially infected nodes, with λ at 2 percent. The results were similar for other networks.

Although high 1-PCI nodes are the most influential spreaders in a single-origin process, all three measures are comparable in this case. The k-shell index is the least effective measure. Node degree is the most effective, closely followed by 1-PCI, but the discrepancy between these values quickly diminishes as the number of multiple original spreaders grows.

2.5 Conclusion

Discovering the most influential spreaders is the key to immunizing complex, dynamic networks against cyberattacks and thereby limiting infection. Overall, μ-PCI, which can be considered a hybrid of node degree and k-shell index, is more effective at identifying influential spreaders and has less computational overhead than either of these traditional measures. Further work could include the use of control-theoretic techniques to improve results.

CHAPTER 3

ACCELERATING SPREADING PROCESSES IN PROBABILISTIC COMPLEX NETWORKS

Influential Spreaders in Complex Networks with Probabilistic Links

3.1 Introduction

In this chapter we propose a centrality measure that can detect influential spreaders in complex networks with probabilistic connections, that is, network connections are associated with a weight value, that corresponds to the common time span connected users spend on their social platforms. Particularly, consider the most popular social networks (SNs), e.g., Facebook or Twitter, where users gain access to the Internet and their social activities through diverse wireless devices (smart-phones, laptops, etc.) and become embedded to the Internet infrastructure swiftly, in different time spans of their everyday lives, to interact, exchange opinions and ideas or simply act like tuners for advertisements. Facebook self-reported statistics note that smart-phone users check online 14 times a day, while an average user spends daily 40 minutes on the site. Now meditate on the vast amount of data traversing through such networks and how this magnitude of information has evolved through time. As reported in [148], in 2007 we had an average of 5000 tweets per day whereas in 2013 we were at 500 million tweets on a daily basis [94], representing

Related publication [**B1**]: Dimitrios Katsaros, Pavlos Basaras. *Detecting Influential Nodes in Complex Networks with Range Probabilistic Control Centrality*, **chapter in Coordination Control of Distributed Systems (Jan H. van Schuppen and Tiziano Villa)**, Lecture Notes in Control and Information Sciences, vol. 456, Springer-Verlag, pp. 265-272, 2015.

Related publication [**B2**]: P. Basaras, D. Katsaros. *Identifying Influential Spreaders in Complex Networks with Probabilistic Links*, **In (Tansel Ozyer, ed.) Social Network and Surveillance for Society**, chapter in book, Springer, accepted, September, 2017.

a six orders of magnitude increase. From the above considerations one could argue on what share of these vast data is actually being 'seen' by its corresponding *audience*, i.e., friends, followers or broadly speaking from the connected society, and on how this is further affected by the different time spans that individuals spend on their social activities.

It is evident that users cannot follow such immense traffic of data, but what of time limited messages or alerts? As an example let's reminisce the *Twitter Faster Than Earthquakes* event. In 2011 of August 23rd, it took 30 seconds for an earthquake to travel from Washington DC to New York, but tweets were fast enough to reach the New York city quicker than half a minute. To account for many such cases, for example of natural disasters, Twitter has launched the *Twitter Alerts: Critical information when you need it most* program in September 2013, for its users to receive reliable information during these times. In this study we emphasize on such *Real Time Data, RTDs*, that need to be 'made known' to the largest possible portion of a social network at a short time interval (i.e., within a few minutes or hours) and on the fact that this particular info will serve no further purpose in larger time spans (e.g., days or weeks). Consider an enterprise announcing a discount of a certain 'hot product' but only for a limited stock or a limited time offer, aiming to attract large masses of consumers. A preeminent question arises; which users should an administrator select as spreading initiators to increase as much as possible the number of potential buyers?

Although we presented the problem in terms of activities over technological social networks, the issue of the effect of concurrent 'activity' is present in other types of complex networks as well, such as human contact networks and their relationship to infectious disease transmission. Theoretically, in such networks a short interaction between a susceptible and an infectious person could lead to a comparable amount of ingested infectious material as that of a long interaction, assuming that the short interaction is more intensive than the long one. However, prolonged contacts tend to be more intensive than short contacts [155].

3.1.1 Motivation and contributions

The issue of identifying influential spreaders in complex networks is a well studied topic that received increased attention in recent years [58], [77], [139]. However for this particular framework of data that we are addressing in the present study, the different patterns in the concurrent activities of 'connected' users will constitute the most essential ingredient for detecting the *Real Time Influential Spreaders*, *RTISs*, rather than simply focusing in a static image of a social network and traditional approaches. At this point we should note that both RTDs and RTISs are connotations to characterize data with relative short lifetimes and influential spreaders for such cases respectively.

Empirical observations [90], [113], [174] note that users in SNs are not active around the clock, and they show a complex behavior and distribution over the time they spend on their social activities. A probabilistic framework that follows such complex behavior could portray the possibility of a link-connection to exist, i.e., when connected users are active, and the dissemination process is on progress. A relative approach is that reported in [90] where the authors illustrate a probabilistic model that accounts for a node-user to be active or not (and thus his connections to be present or not) at the time for example of a disease outbreak or broadly speaking a diffusion process. It is thus an important feature that we need to consider in order to quantify the strength of the corresponding propagation.

Similarly to [90] we model the existence or absence of connections−rather than users−by annotating weights on links that correspond to the *mutual time* that connected users spent on their separate social activities. Intuitively if we could locate those nodes that are the starting points for paths of users which *share at a great degree common time in their online social activities*, it could provide valuable insights to better approximate the spreading capability of users, and thus more efficiently 'control' the spreading process of RTDs. By conducting simulations and experiments in different Social Networks, we will see how the proposed identification technique, namely *ranged Probabilistic Communication Area* ($rPCA$) effectively combined the activity schedules of connected users, identified the most influential spreaders and outperformed the competing techniques in various scenarios.

The present article discusses the issue of detecting influential nodes in complex networks with probabilistic links and makes the following contributions:

- Investigates the issue of detecting real-time influential spreaders by considering the mutual time connected users spend on their online social activities.
- Proposes an adjustable centrality measure, the range Probabilistic Communication Area ($rPCA$) that accounts for such characteristic
- Thoroughly evaluates this centrality measure under diverse competitive techniques in different real networks.

The rest of this article is organized as follows: an overview of related works for the identification of influentials is presented in Section 3.2. Section 3.3 describes the proposed algorithm. In section 3.4 we detail our experimental environment, competing techniques and evaluation criteria. In 3.5 we evaluate the performance of the competitors and finally in 3.6 the conclusions.

3.2 Related Work

The literature on the problems of maximizing the spread of influence and of identifying influential spreaders in complex networks is quite rich during the last decade. In this section we only mention but a few among many important studies. We should also categorize networks depending on the pattern of their connectivity, i.e., directed or undirected networks in order to discuss the direction of the propagation and finally emphasize on directed networks. The first problem was posed in [184] and later investigated further providing more efficient algorithms, e.g., in [96], [165], [183]. Newer approaches to the design of centralities include concepts such as κ-path centrality [93] and distributed algorithms for identifying influentials based on random walks [112]. Other graph-theoretic methods include the k-shell decomposition of a network [139], where the authors discuss that a node's location is an important characteristic for the influence potential of that node. Other approaches based on several shortcomings of k-shell are presented in [58], [97], [105], whereas local techniques that combine effectiveness and efficiency are proposed in [77], [110].

Considering a directed social network, a user i is called a follower of j if there is a directed link from i to j ($i \rightarrow j$), that is, i can receive information from j. Thus for these network cases the diffusion takes place through the incoming connections of a node-user. To detect the most influential spreaders in directed social networks, researchers often employ $PageRank$ [191] centrality, where a node i is considered as influential if it is pointed by many other and important nodes. A variation of PageRank, namely $LeaderRank$ [126], introduced a ground node to the initial network connected to all other nodes through a bidirectional link. LeaderRank outperformed the original algorithm by detecting more efficient spreaders. Finally $weightedLeaderRank$ was presented in [66]. For this approach the authors allow nodes with different in-degrees to get different scores from the ground node. This last variation outperformed its predecessors by identifying more influential spreaders. TwitterRank [149], also a variation of PageRank, was developed for identifying influential spreaders in Twitter. The fundamental difference of the two algorithms is that TwitterRank develops a topic-sensitive random walk, i.e., the transition probability between users in Twitter is topic-dependent. In a way this generates a topic-sensitive network structure, however considering topic specific information in beyond the scope of the current study.

As we mentioned earlier users gain access to their networked environment through diverse wireless devices for arbitrary lengths of time. Such interactions can be projected as temporal networks. Quite often temporal networks are separated in two categories based on time sequences and time intervals for the interactions between connected individuals in communication networks. In our study, we are searching for connected nodes which have common online activity, i.e., they do not necessarily exchange messages at arbitrary times, but rather they are both active in regular times. This can be considered as another simplification of temporal networks were we discuss the probability of existence of interacting paths based on such observations. For more details on temporal networks readers are referred to [54] and references therein.

3.3 Proposed Technique

In this section we present the proposed technique, the *range Probabilistic Communication Area* ($rPCA$).

3.3.1 Complex Networks with Probabilistic Links

A complex network $G(V,E,w)$ is a directed graph where V is the set of vertices (nodes), and E is the set of pairs of vertices (edges). Every edge is described by a weight $w \in [0,1]$ and a direction. Each vertex involves in- and out-neighbors. As usual, the number of head endpoints adjacent to a user-node is called its *inDegree* (k_{in}), and the number of tail endpoints defines the node's *outDegree (k_{out})*. The weight values associated with every edge define a network structure which describes the probability for any two connected nodes to be both active, for example during a diffusion process. As we will see later in our experimentation the mining and efficient use of such information will prove a valuable asset for the spreading of RTDs.

3.3.2 r-Hop User Communication Paths (UCPs)

A user communication path (UCP) on a directed complex network, is a directed path consisting of n individuals and *n-1* connections among them, such as no user appears more than once, e.g., $a \to b \to e \to j$ in Figure 3.1. For simplicity the example network is a Directed Acyclic Graph (DAG). To complete our definition we also need to define the range for such interacting paths, as the number of connections that form it or the hop distance from the initial node, e.g., a to j. For our technique the communication paths emanating from each individual node will define its significance in the network. The weight values on the connections will be used to investigate on the quality of paths through which a user i "sees" the rest of the network in range or in other words *to search for users which share common time in their social activities*. An ideal UCP could be the $a \to x \to z$ path, however, this implies that all these nodes are always connected.

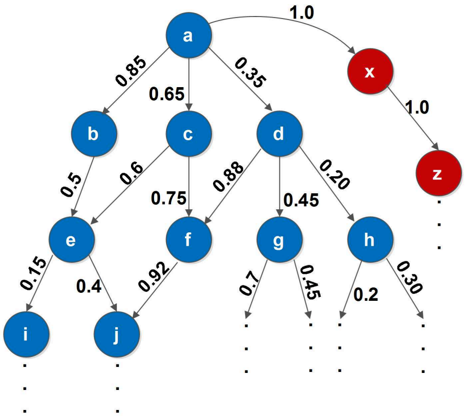


Figure 3.1: $rPCA$ identifies nodes which possess the characteristic that from these nodes emanate "strong" paths. For 2 hops distance: $2PCA(a) = 17.283$ and $2PCA(b) = 1.1$ assuming that both i and j have 2 outgoing neighbors and x,z are hypothetic nodes, i.e., not included.

Finally the strength of those interacting paths needs to be measured with respect to their probabilistic connectivity. To this end, we apply the following formula to measure the strength of an r-hop interacting path ($SUCP^r$):

$$SUCP^r = \sum_{j=1}^{r-1} w_j \cdot w_{j+1} \tag{3.1}$$

where r defines the range of a particular UCP and w_j is the weight value at j hop distance from the originator, i.e., the weight of the corresponding connection. Intuitively if we could rank nodes on the basis of their UCPs, we could potential set the right paths for the spreading of real time data.

Up to this point we presented our proposal for quantifying the strength of a UCP. However how to efficiently combine the weight values associated with the corresponding connections in a communication path and define its significance, is still an open issue. Another formula could be to simply acquire the product of its weights, however such consideration will provide no distinction for paths with relatively equal weight probabilities. For example in Figure 3.1, for the interacting path $a \to b \to e \to i$ we would obtain a value of 0.063. The same value however would be attained if we sorted the weights in any possible way, e.g., by reversing the probabilities of $b \to e$ and $e \to i$ or by placing the weakest interaction first and thus decrease the probability of existence for the path. Another policy could be to assign a measure of importance for a specific weight depending on its hop distance from the originator, i.e., weights closer to the initial node in a UCP are perceived as more vital. However, except for the fact that a tunable parameter would have to be added, the significance of an interacting path like $a \to d \to f \to j$ which starts with a relatively weak weight and henceforth is composed of a strongly connected users, would be belittled with such consideration.

3.3.3 Range Probabilistic Communication Area (rPCA)

Following on these requirements, we built our proposal for defining centrality measures over graphs with probabilistic edges for range-limited neighborhoods. The $rPCA$ value of a node i within a specified range r, is computed as the sum of $SUCP^r$'s emanating from i as follows:

$$rPCA(i) = \sum_{j=1}^{N^*} SUCP^r(j) \tag{3.2}$$

where N^* depicts all different paths emanating from i. Note that nodes quite often share similar vicinities, i.e., they may have a large number of common friends, and thus a certain path may be traversed by more than one ways, e.g., $a \to b \to e \to i$ and $a \to c \to e \to i$. For paths of interaction with hop distance greater than 2, the appearance of cycles, e.g., $i \to j \to k \to j$ is a frequent phenomenon, especially when studying social networks. However considering “cycles of interaction" and thus returning to previous paths (or revisited node regions) is very likely to degrade an algorithms performance and thus these occasions are omitted by definition from our algorithm.

The proposed centrality measure can be defined for both, *the entire network ($*PCA$) and for neighborhoods around each node*. It is within our scope to maintain locality in order to provide an effective and efficient algorithm that can be applied in large scale networks and real time applications, and thus the range of UCPs is limited at relatively low values, i.e., 2 and 3. Generally we could search to any number of hops, however we understand that increasing the range of UCPs beyond the 90-percentile-diameter (cf. Section 3.4.2) will provide little additional information to our approach, since only about 10% of the network nodes remain.

Although we have presented our method considering that information will flow through the out-neighbors of a network node, the implementation of UCPs is straightforward by following the in-links as well, if data flows through the in-connections.

3.4 Performance evaluation

For our evaluation we had to select appropriate competing methods, use networks with probabilistic edges, and also propagation models. In this section we describe our simulation environment.

3.4.1 Competing Techniques

A diverse list of competitors are selected regarding geodesics, the position of a node in the network and approaches based on random walks. A plethora of studies so far use the degree centrality of a node as a baseline method for comparison. Likewise in our experimentation we apply the weighted version of the approach. The *weighted degree centrality* ($wDeg$) of a node i or equivalently the strength of i, is defined as the sum of the weights of the connections incident on i:

$$wDeg(i) = \sum_{j} w_{ji} \tag{3.3}$$

where j depicts the neighbors of i, i.e., those nodes that i can exert influence, and w_{ji} stands for their associated weights.

The farness of a node i is defined as the sum of its shortest distances (d^{w}_{ji})−with respect to the weighted links−to all other nodes of a network. The inverse of farness is noted as the closeness centrality of i. For its weighted implementation ($wClo$), the weights will describe how close or how far connected individuals are to each other as given by the formula:

$$wClo(i) = \sum_{\substack{j=1 \\ j \neq i}}^{N} \frac{1}{d^{w}_{ji}} \tag{3.4}$$

for all N nodes of a network. In our framework $wClo$ aggregates the weights on a shortest path, and thus likewise our approach combines the weight values to provide an alternative technique that measures the strength and probability of existence for those paths.

Shortest path betweenness centrality describes the number of shortest paths for all node pairs (s, t), that use node i as an intermediate. Previous studies [58], [110], [139] found its performance insufficient to measure the spreading power of a node. Here we evaluate its performance in a relatively different environment of weighted interactions and find similar conclusions ($wBet$):

$$wBet(i) = \sum_{s \neq i \neq t} \frac{\sigma_{st}^{w}(i)}{\sigma_{st}^{w}} \tag{3.5}$$

where σ_{st} is total number of shortest paths from s to t and $\sigma_{st}(i)$ depicts the number of those paths that pass through i.

A weighted version of PageRank is also evaluated. The weights are proportional to the probabilities that a random walker will select a particular edge when choosing an outgoing connection from the current node [177]. Therefore, edges with larger weights are assumed to be traversed more often and thus are more important:

$$wPR_i(t+1) = (1-d) + d \cdot \sum_{j=1}^{N} \frac{w_{ji}}{\sum_{l=1}^{N} w_{jl}} wPR_j(t) \tag{3.6}$$

where w_{ji} is the probability of visiting node i from j, when j is an in-neighbor of i, otherwise $w_{ji} = 0$. The damping factor d accounts for random jumps and N stands for the total number of nodes in the network.

Finally we employ the weighted-LeaderRank (wLR) centrality. wLR outperformed PageRank and LeaderRank in several cases [66]. It was proven more tolerant to noisy data, e.g., for scenarios of incomplete knowledge of the network topology. It is variant of LeaderRank, which introduces a "ground" node to the network connected to all nodes:

$$wLR_i(t+1) = \sum_{j=1}^{N+1} \frac{w_{ji}}{\sum_{l=1}^{N+1} w_{jl}} wLR_j(t) \tag{3.7}$$

where w_{ji} is equal to 1 if there is a directed link from j to i and 0 otherwise. If the destination node is the ground node (g) then $w_{jg} = k_{in}^{\alpha}$, where α is a parameter set to 1 in our experimentation.

For the directed and weighted implementation of the majority of our competitors−excluding wLR and $wDeg$−we use the "igraph" R package [1]. igraph considers the weights assigned to each link as costs, i.e., the largest the value the weaker the path. However in our experimentation weights indicate the strength of a link, and thus we invert the original weight values for $wBet$ and $wClo$. A very popular method for the identification of influentials is the k-shell decomposition [139] and its weighted versions, e.g., [83]. However to the best of our knowledge there is no formal definition of the algorithm for directed and weighted networks. Could we have used measures such as $\mu - pci$? To such methods which are based on link counting and coreness, it is not clear how to quantize a "fractional degree" to its integer counterpart. Besides, such a conversion would loose significant part of the information carried by the probabilistic link.

[1] http://igraph.org/r/

3.4.2 Simulation Settings

3.4.2.1 Datasets

Nowadays there is a wealth of real datasets which concern complex networks, however, it is hard to find networks with probabilistic links. Thus, in this article we follow a dual methodology: we work with a real complex network to prove the applicability of our method in a real setting, and four real (initially unweighted) complex networks, in which we annotate their links with probabilities drawn from various distributions. Our simulation setup enables to test the performance of the competing algorithms for scalability, effectiveness and efficiency, across a wide range of networks and link weights.

The real probabilistic network is a contact network measured by the SocioPatterns collaboration[2] using wearable proximity sensors in a primary school, and covers two days of school activity. The sensors detect the face-to-face proximity relations (contacts) of 242 children [53]. The weight of a link is the aggregated contact duration of a pair of children. We normalize the links into the $[0,1]$ interval by dividing each weight with the maximum weight found in the network. The experimental results which concern this real network are presented in subsection 3.5.3.

The procedure for annotating the network links with weights is described in the following lines. We obtained our experimentation networks from the Stanford Network Analysis Platform [65]. For our evaluation purposes the experimented networks were selected based on their connectivity, i.e., three networks with relatively equal number of nodes and decreasing in the number of their respective connections, and finally a significantly smaller network. Specifically, we used the *ego-Twitter* network crawled from public sources, where followers receive information from their followees; *Soc-Epinions1* a who-trust-whom social network of a general consumer review site, where users choose whether or not to trust reviews on products; *soc-Slashdot0922* a technology-related news website, which allows users to tag each other as friends or foes; and finally *Wiki-Vote*, where nodes represent Wikipedia users and a directed edge from node i to node j, represents that user i voted on user j. The base attributes of the aforesaid networks are listed in Table 3.1. The 90-effective-percentile-diameter (90-EPD) denotes the number of edges needed on average to reach 90% of all other nodes.

Network	**Nodes (V)**	**Links (E)**	**diameter**	**90-EPD**	**E/V**	**Type**
ego-Twitter	81,306	1,768,149	7	4,5	21.74	Social
soc-Slashdot0922	82,168	948,464	11	4,7	11.54	Social
soc-Epinions1	75,879	508,837	14	5	6.7	Social
wiki-Vote	7,115	103,689	7	3,8	14.57	Social

Table 3.1: Networks base attributes.

[2] http://www.sociopatterns.org

3.4.2.2 Generation of probabilistic links

For our simulation, the probabilities for the edge weights are assigned based on the Zipfian distribution for a range of skew values $s \in [0.1, 0.9]$. The Zipfian distribution depicts the frequency of occurrence for example of a word randomly chosen from a text, or the population rank of a city randomly chosen from a country. In our framework it will depict the frequency of strong interactions. As s increases we increase in the skewness for the distribution of weights, that is, the strong weights will become more rare. In this study we assume that any two connected nodes would share some common time of networked social activity, but also that there are no identical schedules, i.e., $w \in [0.1, 1)$. The resultant weight values will stand for the mutual time spent by nodes on their online social activities, i.e., will depict the probability of an edge to be present or not at the time of the diffusion process. Links with values close to 1 are mostly active in our inspection time, whereas values near 0.1 are considered mainly inactive. According to these probabilities, we take 10 'snapshots' of the input graph resulting in 10 abstract network images. Similar to [66] to obtain statistically unbiased results, we repeated the computation 100 times for each vertex in each network image, i.e., averages over 1000 spreading processes.

3.4.3 Propagation Model and Influence

As far as the diffusion model is concerned we employ the widely used susceptible-infectious-removed (SIR) model (refer to Appendix A.1). SIR is commonly used for studying the spreading of epidemics in complex networks, where the infected nodes will either get immunity or die [106]. We assume that an interested user propagates "data" only once, i.e., users will not repeatedly send the same information to their respective vicinities. In this study we model the penetration of RTDs in a networked environment, with fixed transmissibility (infection rate) λ for all node pairs. The diffusion process unfolds as follows: in the initial phase all nodes are in the S state except one node in I. An infected node is given a single chance to infect it's susceptible neighbors and succeeds with probability λ. Immediately after and without loss of generality [66] the node enters the R state. The process continues until there are no nodes in the infected state. Similar to [58] given a directed network, the influence of a node i (IF_i), is defined as the number of recovered nodes at the end of the spreading process, when i was the initially infected node. To obtained unbiased results each spreading process is repeated over 1000 times.

3.4.4 Evaluation Criteria

3.4.4.1 Kendall's Correlation (τ)

To evaluate the ranking abilities of each competing method with respect the the actual spreading potential of each node we use the the Kendall's Tau 'b' (see also Appendix A.3) rank correlation coefficient (τ) [196]. It is a statistic used to measure the association between two measured quantities, e.g., (*2PCA*, *IF*). When $\tau = 1$, we have a perfect correlation, indicating that when node

i is ranked before j by some method, i.e., with greater *2PCA*, then its spreading capability is also higher. For $\tau = 0$, the measured entities are considered neutral whereas $\tau = -1$ implies opposite correlation. Generally the closer we get to 1, the better the correlation of the evaluated approach.

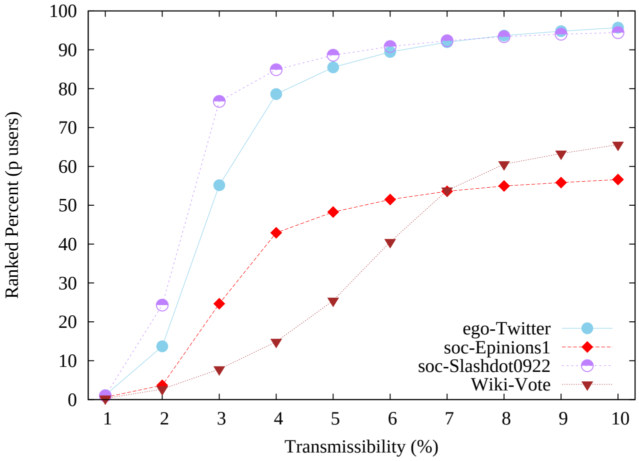


Figure 3.2: Ranked percent with respect to the total number of nodes of each network case for all evaluated λ values, i.e., nodes with $IF > 0$.

3.4.4.2 Fraction of ranked nodes - False Index

As depicted in Figure 3.2 for the lower spreading rates there is a large number of nodes with zero influence, e.g., over 70% for the soc-Slashdot0922 network when $\lambda = 2$. Applying Kendall's correlation to such unfiltered values will provide harsh results. In our simulation we take a closer look for each λ value to provide a more complete assessment and thus the ranked sample used for the ranking process will be composed of user-nodes with $IF > 0$, namely p users. To complete the evaluation of the results and conclude on which technique better identifies the influence power of nodes, we also need to provide an assessment for the rest of the $1 - p$ non-ranked users. The *False Index* depicted in Figures 3.3 to 3.6 (right) fills this void. To obtain the False Index we calculate for each node in $1 - p$ the number of nodes in p whose index is lower from that particular node. In other words we measure the average number of nodes which although did not succeed in propagating, they were ranked with higher index by some users in p, e.g., with greater $2PCA$. Reasonably a small False Index indicates better results.

3.5 Results

3.5.1 Impact of infection probability

In this section we evaluate the efficiency of each competing method in ranking nodes according to their actual spreading potential in four real social networks. For the distribution of links in Figures 3.2 to 3.7, s is set at 0.7. We observe that the most abrupt changes in the curves of correlation for all methods occur at the lower λ values for almost all networks. This is partly because the largest leaps in the percent of the ranked p users occur within the fist few increments of the spreading rate. As illustrated in Figure 3.2, the p nodes constitute about 15% of the network nodes in Twitter when $\lambda = 2$, and about 58% for $\lambda = 3$. The changes in τ however are not only due to the increasing number of the p users used in the ranking process. As the spreading rate increases, the influence of nodes from previous λ values also changes and the same may happen to the ranking between those nodes in subsequent spreading rates.

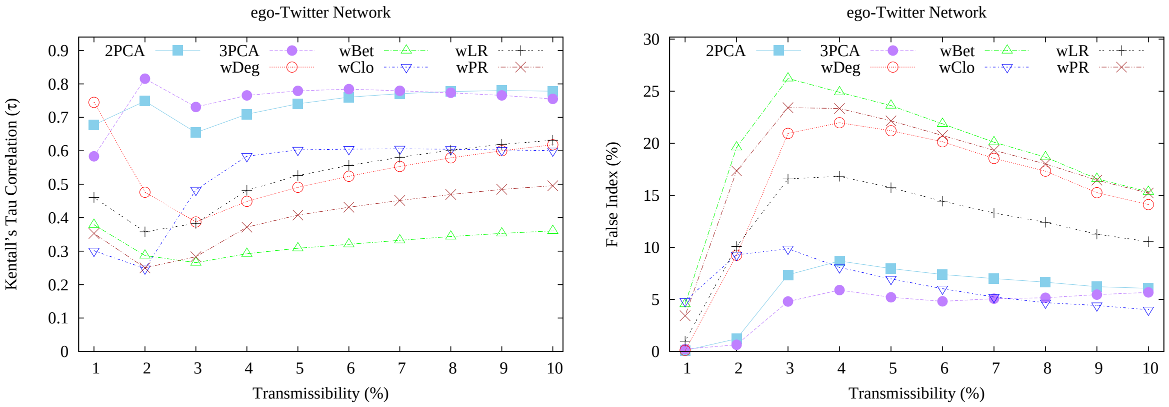


Figure 3.3: In almost all different spreading rates for the ego-Twitter network, the proposed technique significantly outperforms its competitors.

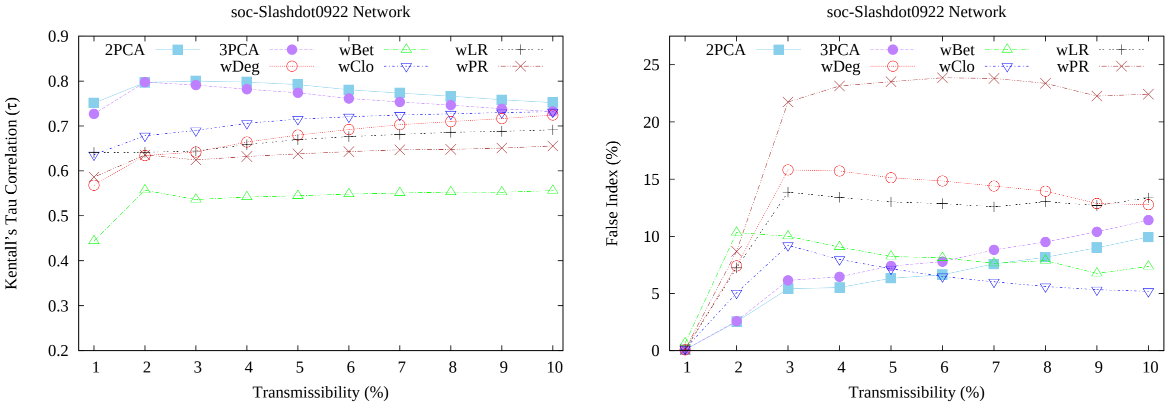


Figure 3.4: For the soc-Slashdot0922 network we observe that our approach coincides with the rest of the competing algorithms only for the higher spreading rates.

Considering the results in Figure 3.3(left), 2-3PCA for $\lambda = 2$, significantly outperforms the rest of the competing techniques. Similar observations can be made for the soc-Slashdot0922 network, i.e., the largest differences in τ are found at the lower spreading rates. For Figures 3.5(left) and 3.6(left) however the above observation does not hold. For these cases we observe a more sedate behavior of the curves as we increase in λ. In Figure 3.4, we observe that $wDeg$ and $wClo$ coincide with our approach when λ is about 9%. It should be emphasized that for very large values of λ, the τ values of correlation for the competitors are bound to crossover and oscillate. This is due to the fact that in such occasions an epidemic will occur regardless of the characteristics of the originator. For the higher spreading rates the true influential nodes are very likely to get infected at some point as the diffusion progresses, and thus result in an epidemic outbreak even though the originator is not truly an influential. Besides by using large λ values the role of individual nodes in the diffusion process will no longer bare significance [58], [97], [105], [139].

When considering the different ranges of our approach we can see that for the low spreading rates there is an oscillation for the most accurate ranking between the two methods. However as we increase in λ for all network cases $2PCA$ always obtained higher τ values, that is, local information of a node's surroundings (communication paths) is more favorable as we increase in the spreading rate.

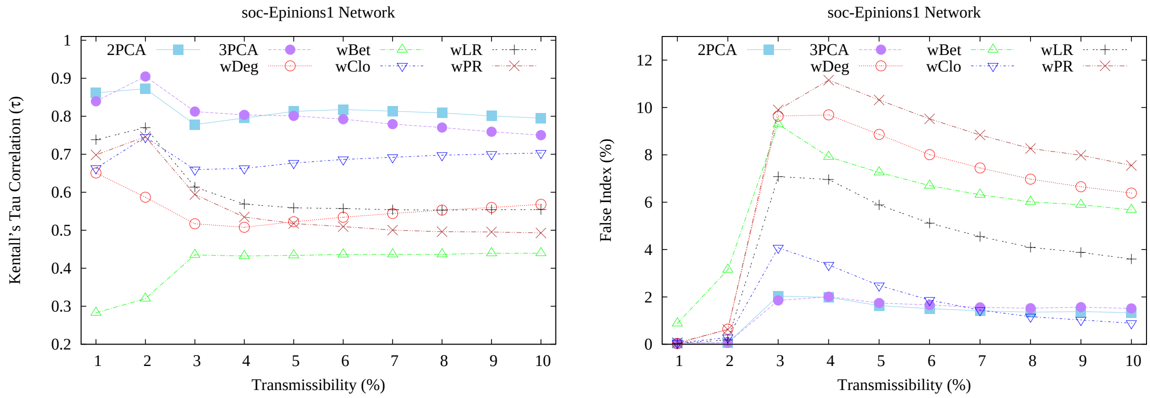


Figure 3.5: As the spreading rate increases, our two-fold approach maintains its superior performance as compared to the rest of the competing techniques.

For an overview on the False Index, 2-3PCA is found at the lower percentages. $wClo$ illustrates similar behavior, however, the rest of the competing techniques illustrate significantly higher values. Note that the False Index does not provide any information about how accurate the ranking for the p nodes is, but rather acts as a further criterion for each respective technique. Ideally we would obtain a zero False Index indicating that all nodes in the $1-p$ set have lower centrality than those in p. Generally a low False Index coupled with a high τ, will promote the most efficient algorithm for the addressed issue. Clearly the proposed technique supports the desired outcome. Only at the higher spreading rates in Figures 3.3(left) and 3.4(left) 2-3PCA illustrates higher False Index.

Focusing separately on each competitor, $wDeg$ is used as a baseline method to illustrate how complete locality serves in quantifying the spreading power of a node. When considering it's False Index we can see that $wDeg$ is rated among the three worst performing methods in all

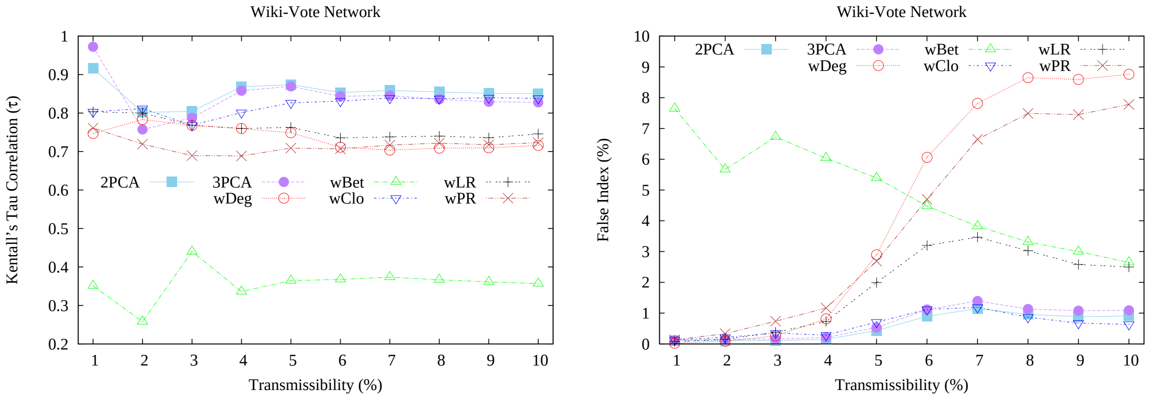


Figure 3.6: For the final network case, an oscillation for the most accurate ranking is observed at the lower spreading rates. Nonetheless, the proposed technique is found within the higher τ values.

evaluated networks. This observation indicates that simply considering the total strength of a node's local connections is not a good indicator to quantify it's spreading influence. For example a high $wDeg$ value may be accumulated by many but otherwise weak interactions, which in our framework is interpreted as regularly absent connections. To our perception such occasions will result in insignificant influence results, and may also be the reason for its high False Index. Furthermore $wDeg$ does not 'carry' any information about the position of the node in a network. Therefore although a node might me connected to it's immediate vicinity with strong links, if it is positioned in the periphery of a network [139], reasonably we expect that its influence will be rather diminished.

Another interesting point seen through our simulation is the performance of wLR. Reminisce that for this particular method no information about the activity schedules is used, and thus we expected a relatively low correlation in our framework of weighted interactions. Although in terms of correlation with influence wLR is outperformed by the proposed technique, when compared to the remaining algorithms we found competitive performance results as illustrated in Figures 3.4(left) or 3.5(left). Generally its performance can be considered relatively similar to $wDeg$'s, however we can conclude that wLR is more efficient, if we consider the False Index of the competitors.

In contrast to wLR, wPR accommodates information from the weighted interactions in the sense that links with higher weights are traversed more often. Both techniques where found to follow approximately the same trend in all evaluated networks as the spreading rate increases, i.e., their illustrated curves either both ascend or descend. However our experimentation showed that wLR obtained both, higher correlation with influence and a significantly lower False Index. Nodes with no outgoing links, the sink nodes, which are indeed present in the evaluated networks, are not well handled by wPR, since they decrease the wPR overall [51]. To our understanding such inefficacy overestimates the spreading power of a node and may be the reason for wPR's low correlation. Generally, through such methods nodes pointed by many other and important nodes are elected as strong influencers, nonetheless, as also noted in [189], [191], quite often the k_{in} of a node is not sufficient to characterize its influence capacity.

Next we investigate on $wBet$ and find that this particular method has the worst performance in all evaluated networks, while other studies [58], [110], [139] also note its inability to capture a node's influence capability. It's lower efficiency can be explained if we consider that through $wBet$, node-users who are unique intermediates for some other nodes (or mediators leading to different communities) are elected as important entities. However in such cases their capability for influence and propagation may well be overestimated if these nodes lead to regions with sparsely connected nodes or small sized communities. In our framework, the problem of identifying influential spreaders is further enhanced by considering the time distribution of nodes social activities. Hence, $wBet$ will be at a further disadvantage if those links correspond to nodes with highly uncommon time spans. Finally the large False Index for $wBet$ further confirms that influence cannot be measured solely through the shortest paths that pass through a node.

$wClo$ utilizes the weighted interactions in the sense that nodes connected through weak links are considered to be relatively far to each other. When compared to 2-3PCA, the competitor is significantly outperformed in the majority of the illustrated results. We attribute its lower performance to the following: first, although the effective diameter for all network cases is relatively small, e.g., between 4.5 and 5, there are still more than 8000 nodes for Twitter and Slashdot0922 networks, and more than 7500 for soc-Epinions within a diameter of 7,11 and 14 hops respectively. However, considering long interacting paths would include a mixed set of connections, i.e., a relatively long path may be composed of both strong and weak links. To this end we expect that techniques that utilize global information of a network's connections to define the significance of a node in the network, will furnish varying results. Figures 3.3 to 3.6 confirm our statement. Lastly unlike our approach, $wClo$ considers a single communication path to all other nodes from the focal node, and in particular the shortest (strongest) paths to those nodes. Nonetheless, rather than a single strong path, it may be more favorable to take into account a number of interacting paths that reach a single user-node, i.e., multiple paths, in our framework of complex networks with probabilistic links.

In Figures 3.4 and 3.6 we observe that $wClo$ coincides with our approach significantly and thus we advance to thoroughly understand the relation of the two methods in Figure 3.7. The spreading rate is set at 10% for both networks where the competitors are closer. The heat values depict the influenced area (IF), i.e., the number of influenced nodes in percent, for paired values of $2PCA$ and $wClo$. For nodes of the same paired values the average IF is used. Note that each axis is normalized to its largest corresponding index. Moreover the outer plots are ranged up to a certain value (e.g. of $2PCA$) which is then resumed in the embedded charts for clarity. From these figures we can further argue that $2PCA$ is the better indicator for the spreading influence of nodes in complex networks with probabilistic links. From the embedded charts we

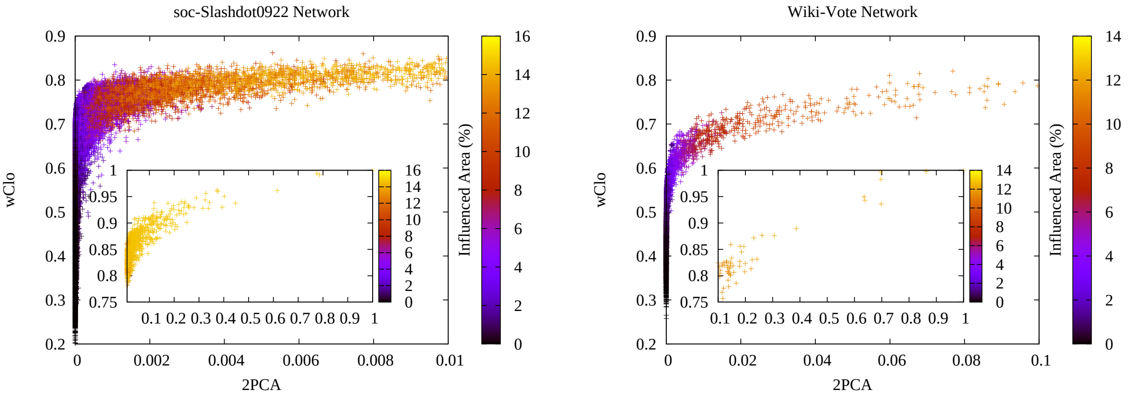


Figure 3.7: $wClo$ was found to coincide with the proposed technique in a few configurations. The presented heat plots, illustrate that influence is closer related with 2PCA. On the contrary, for $wClo$, we observe that the medium values depict an amplitude of influence values.

can understand that the highest index values for both methods indeed correspond to the most influential spreaders. Nonetheless, as illustrated in Figure 3.7(left) between 0.7 to 0.8 of $wClo$, there is a wide variety of influence results, particularly between 4 and 14%, in contrast to $2PCA$ which illustrates a more accurate ranking. We found similar conclusions when comparing $wClo$ to $3PCA$.

Overall, for $rPCA$, paths limited in the near neighborhood of the focal node, i.e., two hop UCPs, are usually sufficient to characterize its role in a spreading process. In our framework, the probabilistic property affects the diffusion dynamics and we thus urge for a technique that effectively handles the different probabilities for connected nodes. Our ranged approach was found quite effective and efficient by better identifying influential spreaders in various networks.

3.5.2 Impact of Zipfian skewness

In this set of experiments we investigate on the skewness of the Zipfian distribution. Due to similar results we present only those for the ego-Twitter network in Figure 3.8. The spreading rate is set at 2%. The percentage of nodes that succeeded in propagating (p users), is illustrated with the colored cycles mapped to the corresponding heat values in the palette. As a first observation we note that as we increase in s for the distribution of links, the number of users that are able to propagate in their respective vicinities decreases. This phenomenon is anticipated as we distance our experiments from uniform distribution and gradually force the weights towards the lower possible values. In our framework such configuration results into frequently absent connections resembling a realistic social environment, where we cannot expect node-users to have largely common time spans for their social activities.

As shown in Figure 3.8 most of the competing techniques illustrate similar behavior in both evaluation criteria, i.e., decreasing and increasing trend for the False Index and τ respectively.

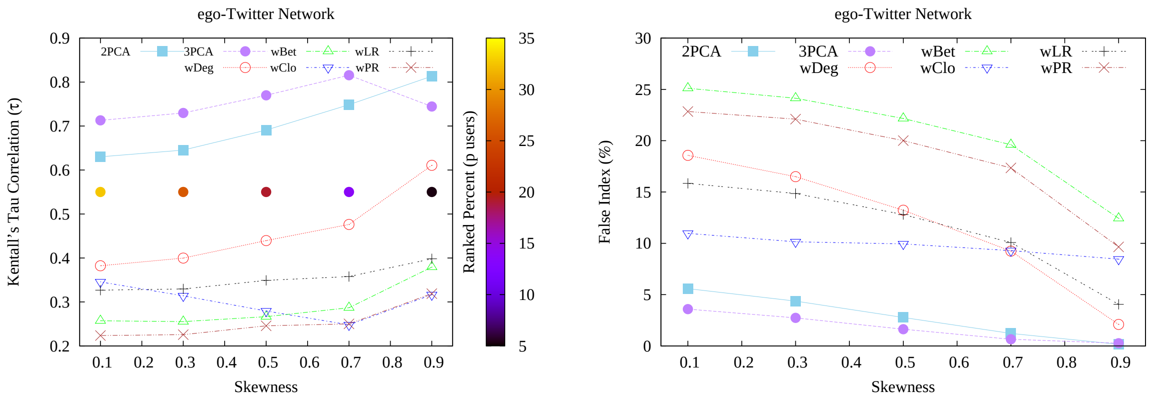


Figure 3.8: Ranging in skewness for the distribution of links. The spreading rate is set at 2%.

For the lower s values we observe only small increases in τ. However as we further increase in s, the changes in τ become more evident. This is due to the fact that for the larger skews, the now fewer strong links and interacting paths become more clear for the competitors. This remark is most visible when $s > 0.7$ where we observe the most significant changes for all methods. wLR however, shows minor changes in τ, an observation somehow coherent with [66] where the authors explain the robustness of the technique in “noisy” networks, i.e., missing links.

When we have a fairly good distribution for the weights (low skewness), we observe that $3PCA$ obtains the highest correlation followed by $2PCA$, whereas the rest of the competing techniques obtain significantly lower values in τ. This observation indicates that when we have many strong interactions, i.e., nodes with highly common activities, accumulating information from relatively long UCPs indeed results in better correlation. In an opposite scenario where node-users have significantly different schedules (large skewness), the strong weights become more rare. Using long paths composed of weak interactions will degrade our algorithms performance which explains the steep fall of $3PCA$ for the higher s values. Conversely thinking we can understand the illustrated behavior of $2PCA$ which uses short ranged communication paths and takes the edge on our ranged approach in the aforesaid cases. The significant difference in the False Index values between the competitors and 2-3PCA further strengthens the superiority of our method. For instance 2-3PCA's “misjudgment” near 0.9 becomes almost zero, whereas in most of the evaluated scenarios (different skews) it is found below 5%. Finally we conclude that in a framework with probabilistic links that portray the property of active nodes as dercribed in our work, considering multiple paths and moreover multiple alternative paths (unlike $wClo$) is a first step for devising an appropriate method for the identification of real time influential nodes.

3.5.3 Evaluation with a real complex network

After the detailed performance evaluation of the methods across a range of network sizes and link weight distributions, we use a real weighted complex network in order to confirm the practicallity of the problem examined and also to further support the superiority of the proposed method.

Recall from subsection 3.4.2.1 that this is a contact network measured by the SocioPatterns collaboration[3] in a primary school. The sensors detect the face-to-face proximity relations (contacts) of 242 children [53]. The resulting network has 242 nodes and 4024 links, after removing the nodes terms as “Teachers" and their interactions, because the network offer no possibility to differentiate between different teachers. Figure 3.9 depicts the number of interactions per pair of children. According to the methodology of data colletion (sensor beaconing) each contact lasts for 20 seconds. Thus, this figure shows in an equivalent way the aggregated contact duration of a pair of children, which is the link weight in our case. Evidently, this distribution follows a power-law, where the majority of the pairs of children have less than 10 contacts.

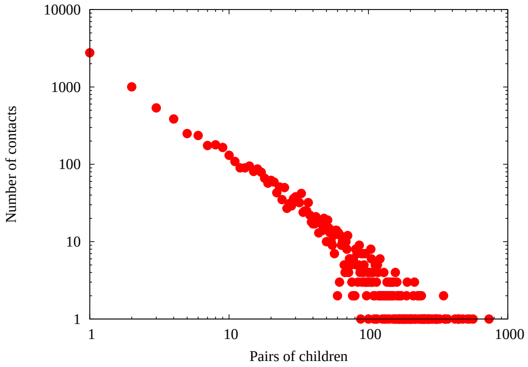


Figure 3.9: Distributions of the link weight (i.e., aggregated contact duration) of the real weighted network.

The evaluation of the competing algorithms is presented in Figure 3.10. The first comment concerns the transmissibility rates in order to achieve high enough infection. The generic comment is that the infrequent student interactions require higher transmissibility rates for successful transitions. Specifically, for the lower λ value, only about 2% of the network is infected, e.g. from an emerging flu originating from the most influential student, whereas when λ= 60, the infected students rise up to 30%.

Regarding the performance of methods, we observe that the best strategy – consistent with our previous resulsts – is $2PCA$, whereas $wBet$ is the worse strategy. The position of the second best performing stratedy is now occupied by $3PCA$, $wDeg$ and WPR (subject to some variation). The interesting thing is that $wClo$ which was steadily the third winner in our earlier finding, now it is fifth. Based on the rankings we obtained for this real network and the conclusions by Figure 3.9 and Figure 3.8, we can say that the link weight distribution of this network is highly skewed for which networks we already have seen that the performance of $3PCA$ and $wClo$ degrade singificantly. Finally, complementary to the False Index illustrated for the artificial networks, we observe (right plot Figure 3.10) no different qualitative results, i.e., the proposed

[3] http://www.sociopatterns.org/2015/01/a-high-resolution-social-network-measured-in-a-primary-school/

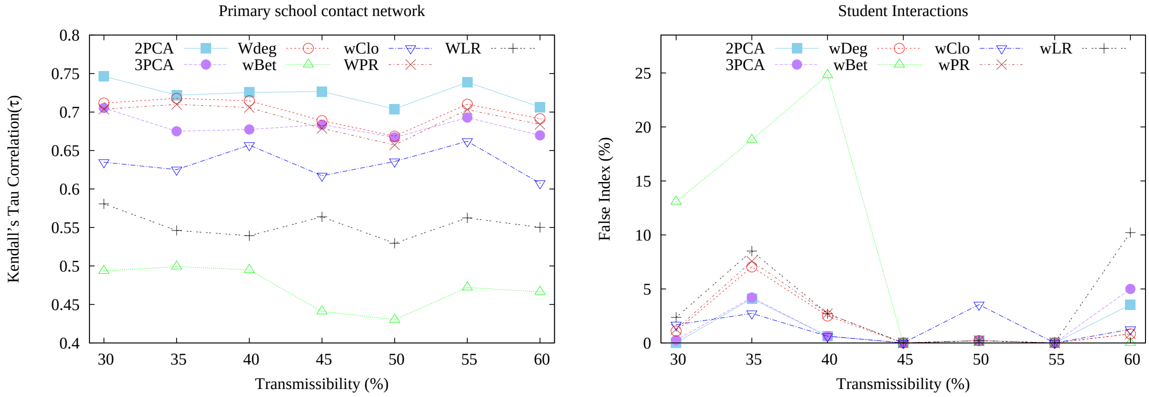


Figure 3.10: Evaluation of competing algorithms over the real weighted network.

technique is found at the lower false values, which further strengthens the superiority of 2-3PCA for the addressed issue.

3.6 Conclusions

The evolution of social networks to date indicate that the amount of information flowing though user interactions is only going to increase. In this article we argued on what portion of information remains 'unseen' from interested users due the continuous flow of data in such networks. With this consideration we focus on 'pieces' of information with limited lifespans, i.e., for data who are interesting to some users but only for a limited time (RTDs). In order to push information into a network and spread RTDs to the largest possible extent we need to account for users which share at a great degree common time in their social activities. With this demand social networks must be remodeled to probabilistic structures. In this study, we used probabilistic links to simulate the probability of connected users with common social activity, and proposed a centrality metric, namely $rPCA$, which accounts for probabilistic communication paths around the focal node. The proposed technique, was evaluated under different spreading rates and distribution for the weight probabilities, and proved superior from its competitors in ranking nodes according to their true spreading potential. Finally, to our understanding, how each method uses-filters the lower weight values is a determinant factor to its performance, since users with low common time-spans will contribute little to each others influence. Moreover in order for RTDs to be substantially propagated we need not only consider the strength of each individual link separately but rather as combined attributes within the interacting paths. For our future direction we intend to apply different approaches for quantifying the strength of the UCPs and further improve our formula for the identification of influential spreaders.

Chapter 4

Accelerating Spreading Processes in Multilayer Complex Networks

Detecting influential spreaders in complex multilayer networks

4.1 Introduction

This chapters focuses in the identification of influential spreaders in complex multilayer networks. So far, the literature on this topic−and the study of complex networks in general−has focused on single-layer networks, where the entities (nodes) and their “communication" channels (links) are assumed to belong to the same network. However, the last few years, we are witnessing a phenomenal initiative in the analysis of new kinds of complex networks, where the interacting entities are assumed to belong to more than one network, called *layers*. These networks are termed multiplex [79], multisliced [141], multilevel [102], interdependent [132] or more general, multilayer networks [49], [61]. Online social networks, financial systems, transportation networks are such networks to name a few; more detailed examples can be found in [49], [61]. Research in the realm of multilayer networks investigates topics such as centralities [38], communities [12], growth models [46] and so on. Similarly, the study of spreading processes in multilayer networks has started to attract significant interest, however the field is still developing its basic principles [45]. On the other hand, the literature on developing algorithms for identifying influential spreaders in multilayer networks is yet very narrowed. However, the spreading of information,

Related publication [**J1**]: Pavlos Basaras, Giorgos Iosifidis, Dimitrios Katsaros, Leandros Tassiulas. *Identifying Influential Spreaders in Complex Multilayer Networks: A centrality perspective*, **IEEE Transactions on Network Science and Engineering**, accepted, October, 2017.

rumors, advertisements, or broadly speaking anything that can be 'shared' through networked populations is rarely isolated into a single network; for instance, information propagation over social networks is taking place in a fashion such that a user decides to share a 'chunk of information' through his/her account, in both Facebook and Twitter.

The identification of influential spreaders in single-layer networks, after the seminal work [139], has concentrated around the idea of 'network decomposition' using concepts such as the k-shell, the k-truss [122], the onion decomposition [18], and so on. All these techniques are iterative and therefore slow; they require knowledge of global network connectivity, in order to locate nodes which are highly connected, hoping that they are also good spreaders. However, all these methods are inapplicable in multilayer networks, because they result in a vector of values for each node [48], i.e., the value of k-shell, or of the k-truss of the node in each layer. Thus, the ranking of nodes using these vectors is not straightforward, unless we define a set of weights for the set of layers and compute a score out of these weights. Apparently, the introduction of artificial weights and computations over them is arbitrary and thus not desirable. An alternative is to address the problem as a "rank aggregation" problem [64], [187], and fuse the ranking lists produced by each value; still, the selection of the fusion algorithm will raise questions about its appropriateness and fairness. On the other hand, the use of centrality measures, such as the shortest-path betweenness centrality, presents the same drawbacks as their counterparts for single-layer networks as analyzed in [77], whereas the use of a PageRank centrality measure adopted for multilayer networks as in [87], has the drawback that its computation requires an artificial ordering among the layers of the complex multilayer network, therefore making this solution to depart from reality. On the other hand, the elegant and mathematically sound generalization of PageRank reported in [29] simply suffers from the computational complexity of the original PageRank, i.e., it is network-wide and iterative, thus time-consuming.

A different line of research on the topic of single-layer influential spreaders detection was described in [77], where the concept of *Power Community Index* (*PCI*) (cf. Definition 1) — and also in [77], [150] — was proposed to detect highly effective spreaders. The proposed method is localized, requiring only local (i.e., two hop) neighborhood information, is fast and proved superior to k-shell. The connectivity of the nodes identified as highly influential spreaders with the aid of PCI is in accordance with the findings of the study [41], which proved analytically that the most effective influential spreaders are those who "...are relatively low-degree nodes surrounded by hierarchical coronas of hubs." In principle, the generalization of the ideas of PCI for multilayer networks would be appropriate, because it would be based on local information of the topology, thus minimizing the computation cost and eliminating the need for having complete knowledge of the entire network state, hence being a good candidate even for real-time applications over massive multilayer complex networks.

This chapter investigates the problem of identifying influential spreaders over complex multilayer networks, by introducing a family of centrality-like measures tailored for local computation only, and able to locate nodes in dense areas of the multilayer network with many intra- and inter-layer links facilitating the rapid evolution of a diffusion process. The chapter makes the following contributions:

- It thoroughly investigates the topic of identifying influential spreaders in multilayer networks by maintaining and exploiting the multilayer structure, i.e., without blending and/or weighting – and thus eliminating – the layers as done by [17] (such an approach has already been proven inadequate and inefficient [29]).

- It proposes a family of localized measures that effectively and efficiently address the problem of influentials identification by incorporating multilayer characteristics (existence and density of intra- and inter-layer connections). The proposed methods can be straightforwardly adapted to any type of multilayer network.

- It evaluates the proposed techniques in a wealth of real and semi-synthetic multilayer networks using as competitors all the major high-performing measures, i.e., PageRank, Betweenness, Degree, k-core and their multilayer variations.

- It concludes that one of the proposed methods, namely $mlPCI$ is (almost) always the best-performing method irrespectively of the size and characteristics of the investigated complex networks, whereas the traditional ones such as PageRank and Betweenness centrality fail to achieve competitive performance.

The remainder of this paper is organized as follows. In section 4.2 we provide formal definitions and notations for multilayer networks. Section 4.3 describes and exemplifies the proposed methods, whereas Section 4.4 outlines the experimentation settings, datasets, competitors and performance measures. In Section 4.5 results are demonstrated, and finally Section 4.6 concludes the article.

4.2 Preliminaries

We are interested in two types of networks, (i) generic multilayer networks, and (ii) multiplex networks. We adopt a graph-theoretic notation and terminology, similar to the one presented in [49]. On the other hand, tensors comprise a similarly powerful, and more compact way to represent multilayer networks; they have been used extensively for the representation of such networks, and for the calculation of centralities and communities in them, e.g., [82], [142]. However, since the measures we introduce in Section 4.3 make use only of local (around each node) information and they can be very easily described with graph-theoretic terms, we prefer to use the graph-theoretic representation. The rest of the section reviews the notation (Table 4.1) of multilayer networks and the spreading model.

Notation	Description
G_i	A monoplex network i
V_i	The set of nodes of the monoplex network i
E_i	The set of edges of the monoplex network i
$\mathscr{P}$	A multilayer network
L	The set of layers of the multilayer network
$\mathscr{G}$	A set of monoplex networks: G_i, $i \in (1,N)$
$\mathscr{E}$	A set of edges between different monoplexes
λ_{ii}	Spreading rate at layer i
λ_{ij}	Spreading rate from layer i to j
k_{in}, k_{out}	in-degree, out-degree

Table 4.1: Notation for multilayer networks.

4.2.1 Monoplex, multiplex and multilayer networks

A *Single* or *Monoplex* network is represented as a graph $G_i(V_i,E_i)$, where V_i is the set of nodes and E_i is the set of edges which connect those nodes. Edges can be directed or undirected, weighted or unweighted. A *multilayer network* can be described as a combination of graphs, G_1, G_2, ..., $G_{|L|}$, and a set of interconnections between nodes in separate graphs. Edges connecting nodes of a single graph are featured as *intra-edges*, whereas edges connecting nodes of different graphs are notated as *inter-edges*. Formally, we describe a multilayer network as $\mathscr{P}(\mathscr{G},\mathscr{E})$, where $\mathscr{G}$ =$\{G_i; i = 1,2,..,|L|\}$ is a set of graphs, i.e., the layers of $\mathscr{P}$, and $\mathscr{E} = \{E_{ij} \subseteq V_i \times V_j;\ i,j\ \in \{1,2,...,|L|\},\ i \neq j\}$ is the set of inter-edges between nodes of different layers, i.e., different graphs. Figure 4.1 depicts a four layer multilayer complex network.

Multiplex networks are a special case of multilayer networks, where nodes are clones (counterparts) of themselves in each layer, i.e., $V_1 = V_2 = ... = V_N = V$. For multiplex networks the only inter-connections allowed are between a node and its counterparts in the remaining layers. Formally, E_{ij} =$\{(v,v);\ v \in V\}$ for all $i,j \in \{1,2,...,N\}$ with $i \neq j$.

4.2.2 Diffusion in multilayer networks

Similar to other studies e.g., [25] we use the *Susceptible-Infectious-Recovered* (SIR) model, which models the penetration of a virus information, product, rumor, etc., in a networked environment (see Apendix A.1). A susceptible (S) node may be a user that is interested in certain information/product. Infectious (I) individuals are those who are already *influenced*, and try to “convince” their susceptible neighbors to follow the same action. Finally, recovered (R) nodes are those nodes who, e.g., have bought the product and can no longer be affected. The diffusion process ends when there are no nodes left in the I state. Hence, *influence* is measured by the number of nodes in the R state at the end of a diffusion process.

In multilayer networks the propagation is expected to diffuse over the different layers at

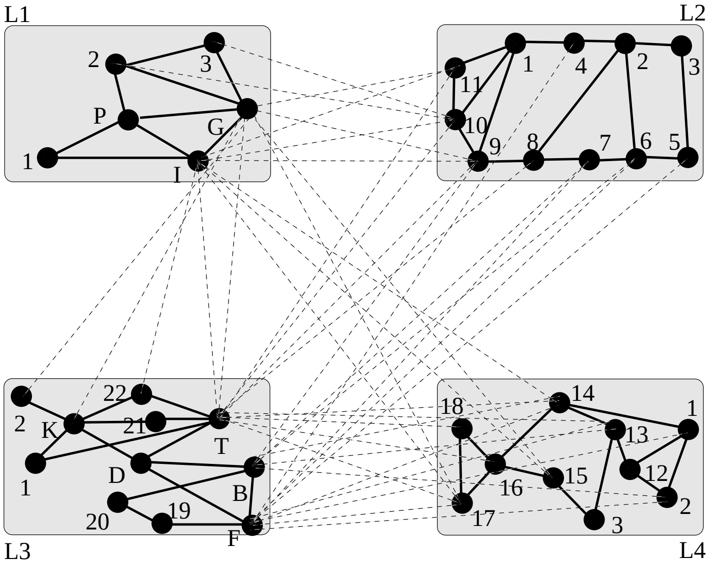


Figure 4.1: A multilayer network consisting of four layers L1, L2, L3 and L4. Nodes with the same ID in different layers depict clones of the same node.

different speeds, i.e., different λ per layer [49], [61]. However the different spreading rates within the various layers is not the only rate that we need to study. Spreading among different layers should also be taken into consideration. Thus, we experience intra-infection probabilities, i.e., infection rate in a single layer i (λ_{ii}), and inter-infection probabilities, i.e., infection rate from a node in layer i to its inter-connection in layer j (λ_{ij}). In multiplex networks nodes are clones in the different layers, hence for this special case $\lambda_{ij} = 1$. In our model, and without loss of generality [58], [77], [139] we assume that an infected source has a single chance to infect its susceptible neighbors, and immediately after it falls to the R state. This is the worst-case scenario to benchmark a method, since the longer a source node is infected the more probable to infect its neighbors. If we allow for (very) long infection periods, then the diffusion process will expand to very large parts of the network (or even to the whole network), irrespectively of the seeding method, the infection probability, the network topology, etc.

Therefore, the question to be answered is *which are those nodes, who if initially activated/incentivised, can trigger a cascade of new adoptions and maximize the spread.*

4.3 Proposed methods to identify highly influential spreaders

Understanding influence in multilayer structures is significantly different from that of monoplex networks; agents (nodes) are subject to different environments which quite naturally have different rules, i.e., ways (paths) to spread information, different spreading rates, etc. Such characteristics introduce new challenges in the domain of influence ranking, and hence new techniques that incorporate those aspects are necessary. In [77] we introduced the *μ-Power*

Community Index (μ-PCI) of a node, that combines the degree of the focal node with the degree of its direct neighbors. The intuition inferred from the understanding that a node in a dense neighborhood, in principle, can affect a large number of other nodes, i.e., exert strong influence. The proposed technique in addition to its local computation cost, successfully identified influential spreaders. Later in [41], it was proved that such connectivity results in the 'best' influential spreaders.

In our current work, we raise and answer the following question: *can we devise a locally-computed measure, that will characterize a node's vicinity, for their density in both, intra and inter connections?* We believe that identifying nodes with strong connectivity in many layers, will reveal potent entities linked to different connected environments, thus able to exert strong influence over the multilayer network. To put our interest into the test, we devise a number of measures that follow our main idea, and evaluate them in a number of real and semi-synthetic multilayer networks.

4.3.1 The family of multilayer PCI measures

For the sake of article's self-completeness, we start with the definition of the original measure, i.e., (μ-PCI), and then give its multilayer generalizations.

Definition 1 (Power Community Index, μ-PCI [150])**.** *The μ-PCI index of a node v is the maximum number k, such that there are at least k neighbors of this node with degree larger than or equal to k in the μ-hop neighborhood of v.*

By setting $\mu = 1$, we get a restricted version of the algorithm, namely PCI. PCI coincides with the well-known h-index [175], and therefore μ-PCI generalizes the h-index for single layer networks. PCI is actually a centrality measure, and it was originally used for the purposes of cooperative caching in wireless ad hoc networks. Later in [77] it has been applied to the identification of influential spreaders; similarly, the h-index has been described as a centrality measure [78], [152] and used in the context of influentials [13], [23].

Next, we provide the generalization of PCI (and thus of the h-index) to multilayer networks.

Definition 2 (Minimal-layers PCI, mlPCI$_n$)**.** *The $mlPCI_n$ index of a node v is the maximum number k, such that there are at least k direct neighbors of v with the number of links towards at least n layers greater than or equal to k.*

From Figure 4.1 with node D as an example: $mlPCI_1(D) = mlPCI_2(D) = mlPCI_3(D) = 3$ and $mlPCI_4(D) = 0$. To combine the distinct n values of $mlPCI_n$ into a single dimension, we propose a simple aggregation. In particular, for a node v we define $mlPCI(v)$ as follows:

$$mlPCI(v) = \sum_n mlPCI_n(v). \tag{4.1}$$

$mlPCI$ by definition bares no strict limitation with regard to either limited, or large number of layers. The indicator will handle cases where nodes are well connected to all layers, to a few or even just one layer accordingly, which indicates the dynamics of Definition 2. According to $mlPCI$ index, nodes well connected in many layers, i.e., nodes assigned high index scores in the range of the n values, will be better "rewarded" from nodes that are well connected, but, in fewer layers. With this understanding we believe that $mlPCI$ will be a good indicator for the spreading potential of nodes.

Simple aggregation can be considered as a baseline method to combine the different values of $mlPCI_n$. However, since larger n implies connection to more layers, a scaling factor could be used with respect to n in order to handle the vector elements differently. Nonetheless, to devise an appropriate method for handling those values is no trivial task. Several factors need to be taken into consideration and further combined with respect to potentially different characteristics introduced by the different layers, e.g., number of nodes, connectivity, global clustering coefficient, etc. Such characteristics can introduce a different view to $mlPCI_n$ and provide a different ranking for the network nodes. In this article we focus on the simple aggregation introduced in Equation 4.1, i.e., agnostic to layer characteristics.

Next, we present a set of special cases of Definition 2.

- *Layer-agnostic PCI* ($laPCI$).
  By ignoring layer information (i.e., ignoring n) in Definition 2, we get a special case of $mlPCI_n$ which we call *Layer-agnostic PCI*, $laPCI$. In Figure 4.1, and considering node D as our focal node, the neighbors that contribute to its $laPCI$ index are nodes K, B, F and T with a total of 6, 9, 12 and 16 links respectively in the different layers. Thus we have four neighbors each of which has at least as many links to the different layers, i.e., $laPCI(D) = 4$. $laPCI$ gives credit to a node whose neighbors have many connections in different layer(s), however, it makes no distinction on how those connections are distributed over those layers. This implies that a node may accumulate a large $laPCI$ value by being well connected in a few layers, and at the same time sparsely connected (or even disconnected) to the remaining ones.

- *All-layers PCI* ($alPCI$).
  We obtain another special case of $mlPCI_n$ by setting n in Definition 2 equal to the number of layers; we call this special case as the *All-layers PCI*, $alPCI$. This approach demands that the neighbors of the focal node have at least k neighbors in *all* layers. Considering node P of Figure 4.1, the neighbors that contribute to its $alPCI$ are nodes G and I each of which has at least two links in all layers, thus $alPCI(P) = 2$. $alPCI$ will detect nodes strongly connected to all layers of the multilayer network that we believe is key ingredient for highlighting the most efficient intra- and inter-layer spreaders. However, this measure will be very restrictive for nodes that lack interconnectivity towards all layers. This may

be a problem for multilayer networks composed of many layers, where it would be quite difficult to detect many nodes with particularly high *alPCI* index.

- *Layer-symmetric PCI* ($lsPCI$).
  Finally, by setting $n = k =$ '*number of layers*' in Definition 2 we get the so-called *Layer-symmetric PCI*, $lsPCI$. This measure is a combination of three aspects: (a) the inter- and intra-degree of the focal node, (b) the inter and intra-connections of its inter- and intra-neighbors, and (c) the layers; all these are nicely "condensed" into a single value. $lsPCI$ alleviates the strictness of $alPCI$: "to all layers" no longer applies, and can be quite effective when dealing with a large number of layers. For limited number of layers we expect $lsPCI$ to act complementary to other methods, since nodes will be ranked from a limited range of values. In Figure 4.1, for node D it applies that $lsPCI(D) = 3$, since nodes B, F and T have at least three links in three layers.

Although we have presented our definitions for undirected networks, their implementation to directed ones is straightforward, i.e., by matching the k attribute to the out-degree of each respective node.

In the next section, we conduct an experimental evaluation of the proposed family of measures providing detailed information about the competitors, the datasets, and the performance measures.

4.4 Evaluation settings

4.4.1 Competitors for multiplex networks

Additive PageRank for multiplex networks (addPR).
PageRank [171] has been used several times for the identification of influential spreaders [70]. In [87] the original PageRank algorithm is extended for multiplex networks requiring though a "predefined" ordering of the layers. We examine here the so-called *additive Multiplex PageRank*, in which the effect of layer i on layer j is exerted by 'adding' some value to the centrality the nodes have in layer j in proportion to the centrality they have in layer i. Since the authors do not provide layer ordering methodology, we order layers in decreasing order of their largest eigenvalue. Our choice is driven with respect to the fact that a larger eigenvalue implies faster information dissemination.

Versatility PageRank (verPR) and Versatility Betweenness Centrality (verBC).
A fundamentally different flavor in extending PageRank for multiplex networks has been described in [29], which, using a tensorial notation, provides a generalization of the original PageRank for multiplex networks, called the Versatility PageRank. Counting the number of shortest paths that pass through a node (i.e., Betweenness centrality) has been widely used as a competing technique for ranking the influence potential of nodes. In [29] the authors generalize

this concept for multiplex networks, describing the Versatility Betweenness. Both techniques are implemented as competitors.

Multiplex k-core percolation methods (Core and sumCore).
We include the k-core percolation for multiplex structures [30], [48] in the competitors lists ($Core$). However, in the evaluation we found only limited values for $Core$. This is due to the fact that $Core$ will follow the coreness of a node's least connected edge type, regardless of how well connected a node may be in the remaining layers. Thus, we also include a variation of $Core$ according to which we calculate the shells for each layer separately and then add those values; we name this version as the $sumCore$ index.

Degree centrality for multiplex networks (aggDeg).
We employ a straightforward interpretation of degree centrality for multiplex networks, i.e., the aggregation of the intra neighbors of the focal node in all layers; we call it $aggDeg$.

4.4.2 Competitors for multilayer networks

The work presented in [50] proposes a generalization of the k-core algorithm that incorporates λ_{ii} and λ_{ij} within the definition of the technique. However, this is not a characteristic that any method should "know" a priori, and hence, we exclude this method from our list. Also, due to the unique characteristic of multiplex networks, i.e., nodes are clones in the different layers, the Additive PageRank (addPR), presented in the previous section, cannot be applied here. Though, we tested the Versatility PageRank and Versatility Betweenness proposed in [29], and $Core$ from [30]. Moreover, in order to provide a complete analysis, we apply the 'traditional' methods, i.e., PageRank, Betweenness centrality, Degree centrality, and k-core by projecting the multilayer network in its aggregated form, implementing in essence the proposals in [17].

4.4.3 Summary of competitors

Table 4.2 summarizes the competitors implemented in this article. Each method's name is comprised by two parts; the latter part discloses the method, e.g., PR stands for 'PageRank', BC stands for 'Betweenness Centrality', Core for 'k-core', 'Deg' for 'Degree', whereas the former part describes the 'flavor' of the method, e.g., 'vers' stands for 'versatility', 'add' stands for 'additive', 'agg' stands for 'aggregated' (i.e., in the aggregated network), 'sum' stands for 'summation' (i.e., summation of values resulting from the calculation of a measure in the different layers).

4.4.4 Datasets

For the evaluation of the competing methods we used several real and synthetic datasets to compare the algorithms in diverse networked environments.

Multiplex networks	Multilayer networks
aggDeg ≡ aggDeg	
addPR [87]	aggPR [171]
verPR [29]	verPR [29]
verBC [29]	verBC [195]
sumCore [this article]	aggCore [139]
Core [30]	Core [30]

Table 4.2: A summary of competing methods evaluated.

4.4.4.1 Real datasets

Table 4.3 depicts the basic attributes of the experimented multiplex networks. For more details, readers are referred to: http://deim.urv.cat/~manlio.dedomenico/data.php. We extracted part of the original networks in such a way that all nodes have counterparts in all layers.

Networks	**N**	**E**	**L**	**Type**	**Nature**
Sacchpomb	875	18214	3	Directed	Biological
Drosophila	1364	7267	2	Directed	Biological
Sacchcere	3096	185849	5	Directed	Biological
Homo	3859	77483	3	Directed	Biological
NYClimateMarch	4150	45334	3	Directed	Twitter
MoscowAthletics	4370	33411	3	Directed	Twitter

Table 4.3: Multiplex networks.

4.4.4.2 Semi-synthetic datasets

For synthesizing artificial networks we follow a similar approach with the authors of [50]. Specifically, we consider real monoplex networks from [65], e.g., several Internet peer-to-peer networks, and synthesize their interconnectivity. Table 4.4 illustrates the real networks used as the different layers of the synthesized multilayer networks. EgV corresponds to the largest eigenvalue of each respective network. We generated two types of multilayer networks: (i) a multilayer network composed of layers with similar size, i.e., *Similar Layers Network* (SLN) and (ii) a multilayer network formed of different-sized layers, i.e., *Different Layers Network* (DLN).

The multilayer network of the first type is composed of the networks/layers (3)–(6) (4 similar-sized layers), whereas the second multilayer network is composed of the networks/layers (1)–(3) (i.e., 3 different layers). For the latter case the different networks differ in the number nodes, edges and network type. We present plots about the out-degree distribution of these networks in the ‘Network properties’ section of the Appendix B.1.

No.	Network	Nodes	Edges	Type	EgV
1.	wiki-Vote	7,115	103,689	social	45.1
2.	cit-HepTh	27,770	352,807	citation	10.8
3.	p2p-Gnutella04	10,876	39,994	p2p	4.4
4.	p2p-Gnutella05	8,846	31,839	p2p	4.3
5.	p2p-Gnutella06	8,717	31,525	p2p	4.7
6.	p2p-Gnutella08	6,301	20,777	p2p	5.1

Table 4.4: Layers of semi-synthetic networks.

4.4.4.3 Generating Interconnections

Since we make use of real networks to represent the layers of the semi-synthetic multilayer structure, we have to decide how to generate the interconnections among layers. We developed a synthetic multilayer network generator which satisfies the following three needs:

- It can define how many interlinks, i.e., inter-neighbors, a node may have.
- It can define how those links are distributed over the layers.
- It can define how links are distributed in each specific layer.

We apply the Zipfian distribution in our interconnectivity generator. The desired skewness is managed by the parameter $s \in (0,1)$. The generator uses one Zipfian distribution per parameter of interest:

- $s_{degree} \in (0,1)$ in order to generate the frequency of appearance of highly interconnected nodes.
- $s_{layer} \in (0,1)$ in order to choose how frequently a specific layer is selected.
- $s_{node} \in (0,1)$ in order to choose how frequently a specific node is selected in a specific layer.

Finally, we need to decide the range of values for the different distributions. For s_{layer} and s_{node} the selection is straightforward since all layers and all nodes within a layer must be available options. Note that the different layers are allowed to have different preferences, i.e., skewness towards different network-layers. Following the review of [61] we understand that inter-connections are rarer than the intra-connections. In our simulations, we limit the inter-degree of nodes within (0, $d \cdot log_2 \sum_i V_i$) for all $i = 1,2,..N$ layers where $d = 1,2,3$ or 4. Hereafter we apply the notation SLN$_d$(s_{degree}, s_{layer}, s_{node}) in order to refer to the generated networks. More algorithmic details and a brief validation of the generator can be found in the Appendix of this paper.

4.4.5 How to evaluate the performance

In our experimentation, in order to evaluate the ranking ability of each competitor, we calculated the correlation of the competitors with respect to the spreading power (SP) of each node (i.e., the number of nodes influenced), when initiating the SIR process from this node as the single origin of the diffusion process. The correlation is measured through Kendall's Tau (τ) "b" (see Appendix A.3) rank correlation coefficient [196]; the τ value between two equi-sized ranked lists is computed as follows:

$$\tau = \frac{n_c - n_d}{n(n-1)/2}, \tag{4.2}$$

where n_c is the number of concordant pairs, n_d is the number of discordant pairs, and the denominator is the total number of pairs of n items in the lists. Some more details are provided in the Appendix. In order to obtain unbiased results, for each node, the average SP is used over 500 SIR processes.

We found that the average is a proper representative for the following reason: we evaluated the ranking ability of the competitors with respect to the standard deviation of the distribution around the average spreading power of each node. In more detail, all competitors were ranked with respect to: (i) the average spreading power (SP), (ii) the average spreading power minus the standard deviation ($SP - std$), and (iii) the average spreading power plus the standard deviation ($SP + std$), when λ_{ii} is the epidemic probability. Hence for each competitor we obtained three values of τ. We found out that these values differ from each other beyond their third decimal point, as shown in Table 4.5, where each cell's value is the ratio between the correlation (τ) of a competitor, e.g., Deg, and the technique which scored the largest τ (i.e., $mlPCI$), when using the respective values of SP for two networks, namely Homo and Sacchpomb. Similar results were observed in the remaining networks, and thus, we draw the correlation of each competitor against the average spreading power.

	Homo			**Sacchpomb**		
	avg-std	avg	avg+std	avg-std	avg	avg+std
aggDeg	0.9839	0.9859	0.9879	0.9899	0.9869	0.9887
sumCore	0.9162	0.9142	0.9112	0.9781	0.9804	0.9806
verBC	0.7020	0.7013	0.7011	0.8020	0.7972	0.8015
addPR	0.8494	0.8457	0.8421	0.8590	0.8649	0.8602
verPR	0.8495	0.8560	0.8529	0.9498	0.9501	0.9530
Core	0.7713	0.7725	0.7717	0.4363	0.4394	0.4340

Table 4.5: Stability of ranking with respect to the average spreading power. The values represent the ratio between the correlation (τ) of a competitor, and the best performing method (i.e., $mlPCI$).

4.4.6 Setting parameters

Table 4.6 illustrates an overview of the experimented parameters, range and default values. In our evaluation in multiplex networks we illustrate how the different spreading rates per layer (λ_{ii}) affect the competing methods. Specifically, we compute the epidemic probability λ_c [118] for each layer, and experiment around this value. For example in Figure 4.2(a), zero in the x-axis sets λ_{ii} of all layers at their respective epidemic thresholds, while -0.2 sets the spreading rate per layer at 20% bellow that value etc. Similar notations are used for the semi-synthetic networks, where we also investigate on the impact of the inter spreading rate (λ_{ij}) and on the density of the generated interconnections (d). To decide the spreading rate between the different layers, we calculate the epidemic threshold of the aggregated network and likewise experiment around this value. We choose to use the same λ_{ij} among all layers in order to give the same "weight" to all interconnections. When evaluating the impact of one parameter, the remaining parameters are set to their default values.

4.5 Results

4.5.1 Ranking influence in real networks

In this section we investigate on the performance of the competing techniques in multiplex networks. For our first and most evident observation we elect $mlPCI$ as the most promising technique for the identification of influential spreaders. As illustrated in Figure 4.2, $mlPCI$ has the strongest correlation with influence in almost all evaluated scenarios, that is, the largest τ. By combining the connectivity that neighboring nodes posses in the different layers as $mlPCI$ suggests, from just one, to all layers of the multiplex network, we show that the proposed algorithm can take advantage of multiplexity more efficiently than the competing techniques.

In plots (d-f) of Figure 4.2, $aggDeg$ performs similarly to $mlPCI$, whereas their in-between performance deviates in (a), (b), and (c). Its worst performance is illustrated in Figure 4.2(b) where the competitor's correlation with influence falls to the fifth place. $aggDeg$ "sees" the network in its aggregated form, i.e., as a monoplex network, and hence disregards a wealth of knowledge regarding the different layers. For instance a node which accumulated most of its $aggDeg$ value from a single layer, is not distinguished from a node of the same index but

Network Type	**Rate**	**Range**	**Default**
Multiplex	$\frac{\lambda_{ii}-\lambda_c}{\lambda_c}$	-0.2 to 0.6	0
Multilayer	$\frac{\lambda_{ii}-\lambda_c}{\lambda_c}$	-0.2 to 0.2	0
	$\frac{\lambda_{ij}-\lambda_c}{\lambda_c}$	-0.3 to 0.3	0
	d	1 to 4	2

Table 4.6: Experimentation parameters.

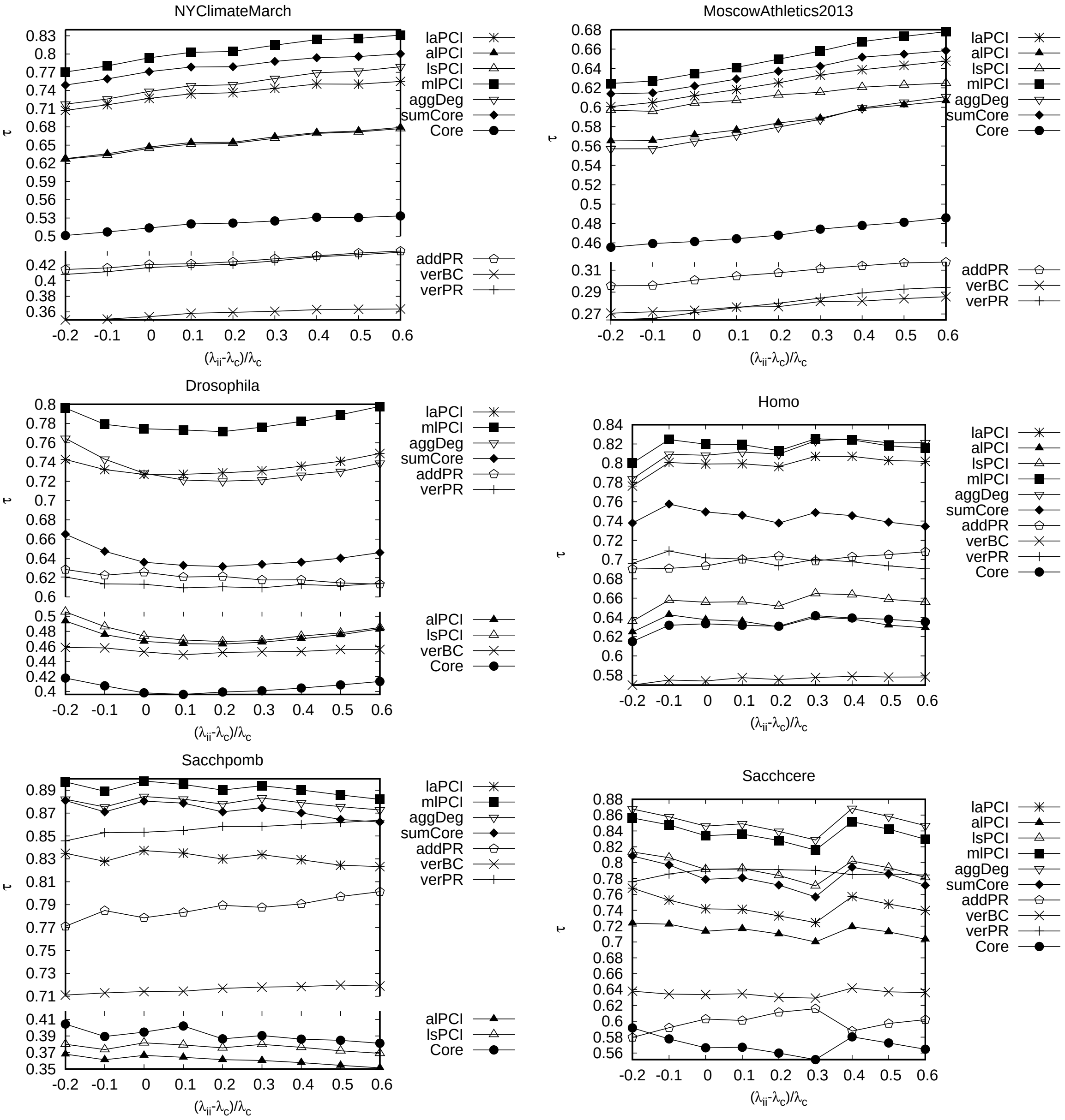


Figure 4.2: Rankings capabilities (Kendall's Tau b) of all competing techniques in real multiplex networks with respect to λ_{ii}. It can be observed that all competing algorithms exhibit similar trends, i.e., either increasing or decreasing trend as the intra-spreading probability changes. $mlPCI$ illustrates the largest correlation with influence in almost all networks. While $mlPCI$ shows a relatively stable behavior, i.e., it is (almost) always at the top of the ranking chain, the remaining algorithms do not posses that property as their rank changes in the different networks, e.g., $aggDeg$ is 2nd in Homo and 6th in MoscowAthletics2013.

equally connected to all layers. Nonetheless, these nodes will have different spreading potential. Moreover, although a node with many connections can be an influential one, it is also a misleading characteristic if the node is positioned in the periphery of the network. This claim has been proven for monoplex networks [139], and it was expected to apply in multiplex structures as well.

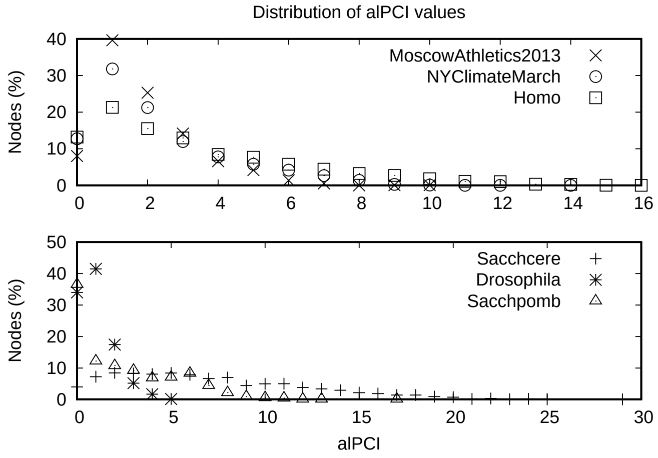


Figure 4.3: Distribution of *alPCI* values for all networks. It can be observed that for most networks the majority of nodes has relatively low *alPCI* values, whereas the largest indexes are appointed to only a few nodes.

Focusing on *alPCI* we observe varying results, i.e., medium performance, as in Figure 4.2(a) or Figure 4.2(b), or low correlation with influence as illustrated in Figure 4.2(c) or Figure 4.2(e). At this point we should reminisce that *alPCI* is a very strict definition which demands connectivity to all layers. Although in terms of spreading capability such characteristic would prove invaluable, in our simulations we found relatively low values for *alPCI*. Figure 4.3 illustrates the distribution of *alPCI* values in the evaluated networks. It can be observed that when we are bound to a poor distribution, i.e., when nodes are not strongly connected to all layers as in Drosophila network (Figure 4.2(c)), we obtain the worst case performance for *alPCI*. Contrary, when nodes are better connected to all layers, the correlation of *alPCI* with influence increases, e.g., as in the Sacchcere network (Figure 4.2(f)). Of particular importance are Drosophila and Sacchpomb networks where we observe a large portion of network nodes with zero *alPCI* index. These are the cases where several nodes act only as receivers (not spreading) in a layer, i.e., zero out-degree. Such instances can be related to lurking behaviors in social networks where nodes only “hear” but never spread information [75]. However, *alPCI* requires spreaders to all layers, hence, by definition these nodes will be “overlooked”. Although the above cases contribute negatively in the evaluation of the proposed mechanism, our results show that finding nodes strongly connected to (as) many layers (as possible) is a key factor for the identification of influential spreaders. For *lsPCI* we also observed variation to its performance. This is due to the relatively low number of layers evaluated (2,3 or 5), and thus limited range of indexes obtained for ranking the multilayer nodes.

Moving to the evaluation of *sumCore*, we observe that the competitor is ranked second in (a) and (b) of Figure 4.2, whereas in Figure 4.2(e) it competes with *aggDeg* for the second place. However, in the remaining networks the competitor performs differently. From Figure 4.4 it can be observed that the largest $sumCore$ values for the Twitter networks are about 10, that is, a large number of nodes distinguished for their influence capabilities from a mere of ten different values (ties are solved via largest $aggDeg$). Although this is a shortcoming shared also by $alPCI$, from Figure 4.4 it can be concluded that as we obtain a better distribution for the $sumCore$ values, that is when nodes are ranked more from their $sumCore$ index than their $aggDeg$, the competitor's performance drops, as it is ranked fourth or lower in our simulations e.g., Figure 4.2(d) or Figure 4.2(f). However, this is an opposite behavior from what we observed for $alPCI$, thus, $sumCore$ cannot be considered a strong indicator for the spreading potential of a node. Furthermore, a relatively poor performance can be observed for $Core$, which can be explained by the fact that the competitor follows the coreness of a node's least connected edge type, regardless of how well connected this node might be in the remaining layers. This characteristic has a negative impact in performance of the technique.

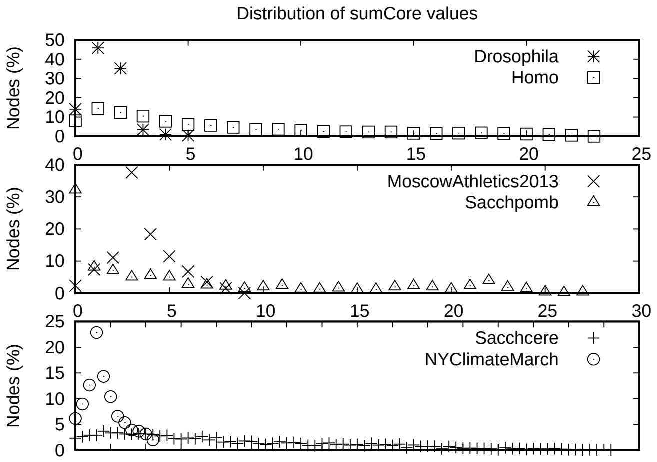


Figure 4.4: Distribution of $sumCore$ values for all networks. According to the illustrated distributions, we observe two groups: (Drosophila, MoscowAthletics2013, NYClimateMarch) and (Homo, Sacchpomb, Sacchcere).

$verPR$ shows an interesting performance. In Figure 4.2(d,e,f) the technique illustrates a very competitive behavior, i.e., is ranked as 3^{rd} or 4^{th} best method in the ranking chain of the competitors; however, in Figures 4.2(a,b) its performance drops. This observation can be attributed to the change in the distribution of in-out neighbors; when these quantities are positively correlated (see Figure B.3 in the Appendix), then $verPR$ exhibits very good performance. When compared to the $addPR$, $verPR$'s performance is either similar or significantly higher, e.g., Figure 4.2(d) and Figure 4.2(f) respectively. This observation concludes that $verPR$ can identify more effective spreaders than $addPR$.

By definition $addPR$ instructs an ordering of layers where a node gains more centrality in a layer if it is important in previous ones, regardless of the node's ability to attract important nodes in the current layer. Although such attribute can be beneficial for a node when it lacks centrality in a layer, but, is well connected in others, it is also a very restrictive characteristic that requires an optimal selection for the sequence of layers, i.e., the order that layers are being processed, overall, should be beneficial to all nodes of a network. Nonetheless, the decision for such ordering is no trivial task especially as the size (in nodes) and the number of layers increases. But apart from this shortcoming, its relative low performance is explained by the nature of the original PageRank when used for influential detection, which assumes that content spreads randomly in the network that is not valid [70].

$verBC$ inherits the weaknesses of the original betweenness algorithm. As an example, consider a node which is unique for reaching a portion of network nodes in a certain area. Clearly, that node will be part of many shortest paths, hence, it will accumulate a large $verBC$ score. However, if spreading in this area is unfavorable, e.g., nodes are sparsely connected, or the target area reached by this unique node is relatively small, the spreading power of that node will not justify its high $verBC$ score in the ranking process. On the other hand nodes that do not reside in any shortest path will acquire a zero index of $verBC$. Nonetheless such nodes may be (directly) connected to hubs, and thus "indirectly" affect a significant number of network nodes. It is straightforward that in such occasions the performance of the competitor will be negatively affected.

Evidently, the competing algorithms will not be equally influenced from network characteristics, i.e., methods that require global knowledge of the network topology are more depended to network topology than local approaches. For instance, by definition, $verPR$, $addPR$ and $verBC$ will be significantly more influenced than the rest of the competing techniques from the distribution of in-out degree (it is illustrated in Figure B.3 in the Appendix), especially when a large number of nodes with low values in either k_{in} or k_{out} are present. To our understanding such characteristics also contribute to their overall significantly lower performance. This is yet another reason for selecting methods that require only local knowledge of the network topology.

Examining the curves of the illustrated results, we observe similar trends for the competing methods, i.e., either increasing or decreasing within a specific range of λ_{ii} values. The observed abrupt changes in τ, as illustrated for example in Figure 4.2(f) for the Sacchcere network from 0.3 to 0.4, or in Figure 4.2(d), is due to a significant amount of newly influenced nodes with respect to those from the previous λ_{ii} value. In contrast to monoplex networks where spreading is of single dimension, in multiplex networks a node can become influenced because it's counterpart was "reached" in another layer. In other words, although there is an influence rate λ_{ii} per layer, the actual spreading rate can be significantly higher when accounting for multiplexity.

Our evaluation so far strengthens our belief for finding influential spreaders in multilayer networks, by imprinting within the proposed measures the density of inter- and intra-connections in the immediate vicinity of the focal node. $mlPCI$, combines those k neighbors connected in just one layer, those k neighbors residing in two layers and so on up to those k neighbors connected to all layers. It alleviates the shortcoming introduced by $alPCI$ and at the same time can be as restrictive as an application requires by setting our focus to at least as many layers as necessary. In addition to its local computation complexity, $mlPCI$ illustrated the largest correlation with influence in almost all evaluated networks for all respective spreading rates, and is thus our primary selection.

4.5.2 Ranking influence in semi-synthetic networks

4.5.2.1 Interconnections and influenced nodes

We start our evaluation by noting the different “rules” that apply for these type of networks with respect to the multiplex structures. First, there are no counterpart nodes, i.e., nodes are different entities, which means that there exists a spreading probability in order to reach nodes in other layers, i.e., λ_{ij}. Furthermore, successfully propagating over an inter-link, only affects one node at one specific layer and not all layers of the multilayer network as in the previous evaluation. The above considerations indicate that we are bound to a significantly different environment, hence, we expect to encounter different results.

Firstly, we examine the effect of the generated interconnections in the diffusion process. We should note that although our generator gives a particular trend on how interconnections are distributed over the layers, the topological characteristics of an inter-neighbor will also play a vital role in the diffusion process. Specifically, an interconnection to a node which resides within a well connected neighborhood will favor the spreading process, whereas the opposite will occur if interlinks are "wasted" over nodes with poor inter/intra connectivity. Figure 4.5 illustrates the cascade size per layer in several networks, i.e., the influence exerted by any initially infected node falls within the illustrated range. It can be observed that the way in which interconnections are distributed over the layers plays a major part in the SIR dynamics; as anticipated for SLN_2(0.3,0.3,0.3) and DLN_2(0.3,0.3,0.3) networks, the cascade size is significantly higher. This is due to the fact that there is no excessive skewness for the inter-degree assigned to the participating nodes (s_{degree}), nor towards which layer those interconnections are guided (s_{layer}), or to the selection of nodes within the target layer (s_{node}). Such configuration will provide a favorable environment for the spreading process, and thus influence a larger portion of network nodes. The opposite scenario is illustrated for SLN_2(0.8,0.8,0.8) and DLN_2(0.8,0.8,0.8). Similarly, having similar distribution for the inter-degree of nodes, e.g., by setting s_{degree} at 0.3 (or 0.8), and vary in the remaining parameters, shows that increased skewness has negative effect on the percentage of influenced nodes.

4.5.2.2 Impact of inter connections and intra diffusion probability

Figure 4.6 illustrates the performance of the competitors in the semi-synthesized networks when evaluating the impact of λ_{ii}. In coherence with our conclusions in real networks, we elect

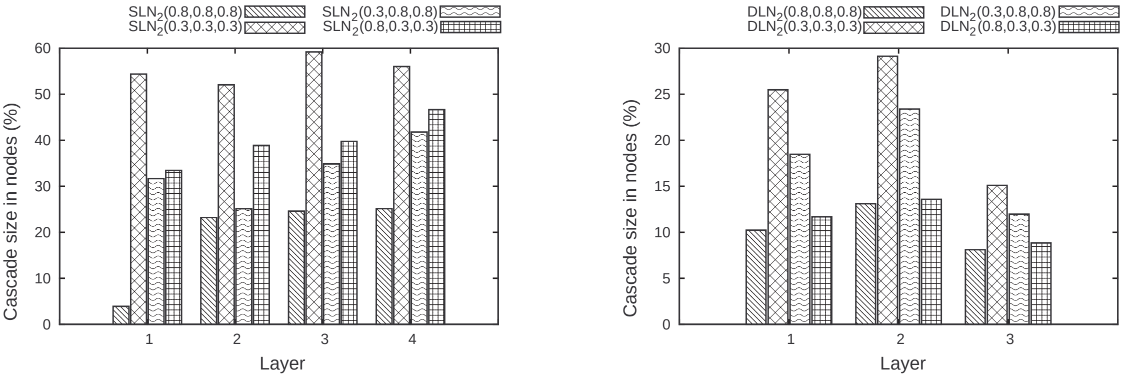


Figure 4.5: Maximum cascade size per layer subject to the distribution of interconnections. It can be observed that when all parameters are set to 0.3 the cascade size is maximum, while the opposite occurs, when all parameters are set to 0.8.

$mlPCI$ as the most promising technique for measuring influence in multilayer networks. It can be observed that $mlPCI$ is at the higher values of τ for almost all spreading rates, however, the ordering for the remaining techniques has changed. Specifically, $laPCI$ can be considered as the second best method, performing almost as good as $mlPCI$ in Figures 4.6(a), 4.6(g) or 4.6(h), and as the next best solution in the remaining networks. $laPCI$ implies k neighbors towards any layers, however, these nodes may reside in many, or, in just one layer. For occasions where the latter holds, and nodes are assigned a large $laPCI$ index, there is strong possibility that an epidemic will arise in the multilayer network, since within these k neighbors, nodes connected to different layers are likely to exist. The same logic applies to nodes with a large $aggDeg$ index as for example in the Wiki-Vote network (details in Figure B.4 in the Appendix). The difference between the two measures that discriminates the performance of $laPCI$, is that those k neighbors that form the node's index, is the result of "filtering" that is applied in the focal node's vicinity, that discriminate a highly connected node within a strongly connected neighborhood, from nodes residing in sparser vicinities. This inherent characteristic governs all proposed methods, which in our view enables the proposed techniques to detect more efficient spreaders.

Of particular importance is the performance of $verPR$ in the SLN networks. Apart from the fact that it has increased correlation with influence with respect to its performance in the DLN networks, Figure 4.6(b) illustrates an interesting result, i.e., $verPR$ outperforms $mlPCI$ when λ_{ii} is larger than the epidemic probability. To explain this behavior we need to consider the distribution of the inter-connections. By setting s_{node} at 0.8, we "send" many interconnections to a certain portion of network nodes within the corresponding layer, that is, in terms of $verPR$, specific nodes are inter-pointed by many others. These nodes will accumulate a large $verPR$ index due to their interconnections, thus rendered as efficient cross-layer spreaders detected by $verPR$. It is due to this intrinsic characteristic of the competitor that we observe its efficient ranking in these specific networks. In Figure 4.6(d), where interconnections are more constrained with s_{degree} set at 0.8, $verPR$ does not outperform the proposed methods, however, as λ_{ii} increases the distance in their performance decreases. In the DLN networks the performance of $verPR$ is far inferior to all methods with the exception of $verBC$. This gap in performance from the demonstrated results in Figures 4.6(a) to 4.6(d), can be explained by comparing the k_{out} distribution of inter and intra links in each respective network type, i.e., in the latter, there is significant difference in magnitude between the inter and intra neighbors. Evidently from Figures B.2 and B.4 (see the Appendix), the impact of interconnections in the DLN examples will be considerably smoother, which explains the behavior of the competitor.

In all the experiments concerning multilayer networks, $Core$ seems (almost) uncorrelated to the spreading power of nodes (i.e., almost zero τ). This behavior is explained directly by the definition of the algorithm; nodes would get a $Core$ value different than one, only if they have connections to all layers. This happens only for very few cases in our generated networks, and thus practically all nodes get the same $Core$ value. This results in the phenomenon that we observe. By examining the performance of $aggCore$ we observe varying results, i.e., below the 5^{th} place in the ranking chain of the competitors, e.g., 6^{th} in Figure 4.6(e) and 10^{th} in 4.6(c). Nonetheless we cannot expect $aggCore$ to be a challenging competitor since it projects all layers in a single dimension and thus neglects the layered structure of the network.

For $verBC$ it is straightforward that the shortcomings discussed in the previous section also apply in the current framework. Generally, when there are fewer paths to the different layers (s_{degree} = 0.8), the limited shortest paths work in favor of the competitor that shows a relative increase in performance, e.g., comparing Figures 4.6(a) and 4.6(c). However, if either s_{node} or s_{layer} is set to 0.8 we observe decrease in τ as illustrated from Figure 4.6(a) to 4.6(b). It can be concluded that we cannot accurately distinguish the spreading power of nodes by counting the number of shortest paths that pass through them. As described in [29], the performance of $aggPR$ ($aggBC$) coincides with that of $verPR$ ($verBC$).

Setting s_{layer} at 0.8, denotes a possible preference developed between the layers, that is, most interconnections are "guided" from a layer to a specific other(s), while the remaining layers acquire limited inter-links from that particular layer. As illustrated from the results, this parameter has a soft impact to all competitors with the exception of $alPCI$. Particularly in Figures 4.6(b) and 4.6(d), due to this setting, $alPCI$ failed to provide an acceptable ranking, since a significant portion of network nodes were assigned zero $alPCI$, or in other words nodes were not inter-linked to *all* layers. In the DLN networks, although less nodes where assigned a zero value, still, the obtained indexes where significantly low and overlapping. For example in DLN_2(0.8,0.8,0.8), most $alPCI$ values were below 6. In these scenarios we can understand the reasons for its questionable performance, however, $alPCI$ can still operate in one more way, i.e., as an additive rank rather than a solo ranking method. This aspect can be related to Figures 4.6(f) and 4.6(h) where a limited range of $alPCI$ values (ties are solved via the largest $aggDeg$) rank a large number of network nodes, or in other words, nodes are ranked more from their $aggDeg$ index than from their $alPCI$. Such combination, results in distinguishing highly connected nodes that have interlinks to all layers, from those that do not posses that property. $lsPCI$ operates similarly to $alPCI$ since its indexes are limited by the number of layers. Thus, the results illustrated in Figure 4.6 is the outcome of ranking nodes according to $lsPCI$, while breaking ties via the largest $aggDeg$ index. Nonetheless we expect that for multilayer networks composed of more layers, $lsPCI$'s efficiency will be distinguished further.

Typically, as λ_{ii} increases above the epidemic probability, the identification of influential spreaders becomes more difficult for any algorithm to detect. This is due to the fact that for large λ_{ii} values, that is, as λ_{ii} deviates significantly from the epidemic probability, an epidemic occurs regardless of the characteristics of the initially infected node [106]. Even if the initially infected node is not an influential one, at broad spreading rates there is high possibility that an influential will be "reached" as the spreading progresses, and thus result in epidemic propagation. Hence true conclusion can only be drawn near the epidemic probability.

It is straightforward to understand that the way interconnections are distributed over the different layers, and to the nodes within those layers, plays a vital role in the diffusion dynamics, and thus, in the performance of the competitors. Hence, for any algorithm in order to be characterized as an efficient technique for the detection of those powerful spreaders, intra and inter connections must be incorporated and combined in the most efficient of ways in order to predict the probability of an epidemic outbreak. Robustness to either limited or increased number of inter-links is also a necessity. Furthermore, it can be concluded that traditional approaches that project the multilayer network to a single dimension cannot predict the actual spreading power of nodes in these complex structures.

4.5.2.3 Impact of inter connections and inter diffusion probability

In Figure 4.7, we investigate on how the competitors behave in the increase of the inter-layer spreading probability. To this end we choose to have a favorable distribution regarding the inter-degree of nodes, i.e., $s_{degree} = 0.3$. First, the ranking obtained from the previous section has remained relatively unchanged. This observation strengthens the evaluation of $mlPCI$ which illustrates a robust behavior to the different spreading rates used in our simulations. Examining the trends of the illustrated curves, it can be observed that all competing methods become more effective as λ_{ij} increases. Focusing on Figures 4.7(a) to 4.7(c), we observe that as λ_{ij} increases above the epidemic probability, the distance in performance of $aggDeg$ with $mlPCI$ and $laPCI$ starts to decrease, and coincides at 0.3. However, similarly to our previous discussion, true influential spreaders can only emerge near the epidemic threshold, where we observe that $mlPCI$ has the largest τ compared to the remaining techniques.

In Figures 4.7(d) to 4.7(f), the performance of $mlPCI$ is distinct even when λ_{ij} is above the epidemic probability. The basic difference between these networks and those in Figures 4.7(a) to 4.7(c) lies in the distribution of inter-intra k_{out} (Figures B.2 and B.4 in the Appendix). Specifically, in the DLN networks, nodes are much more intra-connected in their focal layer than inter-connected to different layers while for the SLN networks, intra and inter connections are more comparable. Hence, for the DLN networks, the inter-connections will have a smoother impact on the spreading dynamics.

Our evaluation so far illustrates that the interplay between the different layers affects the competing algorithms differently. For instance s_{node} at 0.8 affects the performance of $verPR$ positively —also illustrated in the previous section— as depicted for example in Figures 4.7(b) and 4.7(c). $verBC$'s performance decreases when either s_{layer} or s_{node} is set to 0.8, and in fact it is lower, when both parameters are set at 0.8. This observation is most evident in Figures 4.7(a) and 4.7(c).

Similarly to Figure 4.6(b), due to $s_{layer} = 0.8$, $alPCI$ is unable to rank nodes in the SLN$_2$(0.3, 0.8, 0.8) network (Figure 4.7(c)). This is due to the fact that nodes are not interconnected towards all layers. Nonetheless, from Figures 4.7(d) to 4.7(f), we can observe that even when $alPCI$ ranks nodes with a limited number of different indexes, by breaking ties via the largest $aggDeg$ policy, we obtain a significant improvement in τ.

The above considerations are vital ingredients for building a successful recipe that will detect influential nodes in multilayer networks. It is our belief that all these characteristics must be imprinted within a technique in hopes of understanding and predicting the spreading power of nodes. $mlPCI$ inherently filters a node's near vicinity, i.e., *those "k" neighbors at least "k" connected from just one to all layers of the multilayer network*, which as shown in the majority of the illustrated results, separates it from the rest of the competing algorithms.

4.5.2.4 Impact of increasing interconnections (d)

Our final section illustrates the performance of the competitors as we increase in the density of interconnections (see Figures 4.8 and 4.9). Reminisce that all spreading rates are set to the epidemic probability, however, as d increases, the epidemic probability of the aggregated network decreases, i.e., λ_{ij} decreases. This observation is evident in the SLN networks even at the initial values of d. For instance when s_{degree} is set to 0.3, the largest eigenvalue is about 10, 15, 21 and 27 for $d = 1,2,3$ and 4 respectively. Evidently, the increase of the largest eigenvalue, and thus the decrease of the epidemic probability, is confoundedly significant. For $s_{degree} = 0.8$, we observe a smaller increase, e.g., 8.5 for $d = 2$, however such behavior is anticipated due to the distribution of inter-connections. In [118] the authors state that the epidemic probability of the aggregated network is smaller than that of the individual layers. This observation is coherent with our study in the SLN networks (Table 4.4), however, for the DLN case, where the multilayer network is composed of layers with different number of nodes, edges, degree distribution etc., we found that for $d \leq 2$ the epidemic probability followed the eigenvalue of Wiki-Vote (about 45), that is, the layer with the largest eigenvalue. Even when we increased d up to 4, we did not observe a significant increase, e.g., 47 and 49 for $d = 3$ and 4 respectively. This is due to the large difference in the distribution of k_{out} of the inter and intra neighbors.

In particular, examining Figures 4.8 and 4.9, we observe similar results with our previous discussions. Evidently, the algorithms perform differently in the SLN networks with regard to their performance in the DLN scenarios. The former depicts a decreasing correlation with influence as d increases, with the exception of $verPR$, whereas the latter shows a more complex behavior. At this point we should note that in the SLN networks, the increase of d employs a growing number of interconnections that surpass that of the intra-links for $d > 2$. In terms of $alPCI$, this attribute is not advantageous, since nodes will be indexed for their k neighbors to all layers, thus their rank is bounded to the limits of their intra-connections. On the contrary in the DLN networks which are not governed by such rule, $alPCI$ has increased correlation with influence, performing similar to $mlPCI$ when $d \geq 3$, i.e., when nodes have more connections to all layers.

4.6 Conclusion

Multilayer complex networks have recently been the focus of intense study in the realm of network science. Real instances of them include transportation networks, online social networks, power networks and so on. Diffusion processes, such as spreading processes, cascading failures, cooperative behavior are significant fields of study. Among them, the identification of influential spreaders is a significant task due to its application in immunization strategies, advertising and so on.

This article investigated the problem of identifying influential spreaders over multilayer complex networks, since we are currently 'embedded' in multiple networks concurrently, e.g., in the case of online networks, we have an account at Facebook, Linkedin, Twitter, etc. and we spread our ideas/product-preferences using all of them. The article explained the lack of proposals so far for carrying out this task, and explained the inadequacy of the corresponding techniques proposed for the same problem in the case of single-layer complex networks because they do not take into account the existence of multiple layers and/or generate solutions that do not allow the straightforward ranking of nodes for selecting the most influentials.

Then, it proposed a family of measures for describing the strategic position of a node within a multilayer network. These measures condense into a single number the connectivity of the node with respect to nodes belonging to the same layer as well as to the rest of the layers. The calculation of these measures requires only information of the connectivity of the surrounding nodes, and not iterative computations with knowledge of the network-wide topology thus making it scalable, and quickly computable. Moreover, this feature makes them suitable both for online (e.g., response to evolving infections) as well as offline mining tasks (e.g., selection of best 'promoters'), due to the huge size of underlying networks.

The experimental evaluation of the proposed methods carried out against all major competitors proposed so far for either single-layer or multilayer networks, i.e., degree, betweenness centrality, PageRank and k-core for single and multilayer/multiplex networks. The complex networks used for the evaluation spanned a wide variety of network structure and size, and a network generator was also developed and used so as to test a wide range of topology characteristics. The final outcome of the evaluation marked $mlPCI$ as the best performing measure for almost each and every dataset used. Its success can be attributed on building on the shortcomings and embedding the benefits of the members of its family proposed in this article; it achieved to summarize the connectivity around a node in a concise and quite accurate way, even though it refrains from examining the whole network topology with time-consuming iterative decomposition procedures.

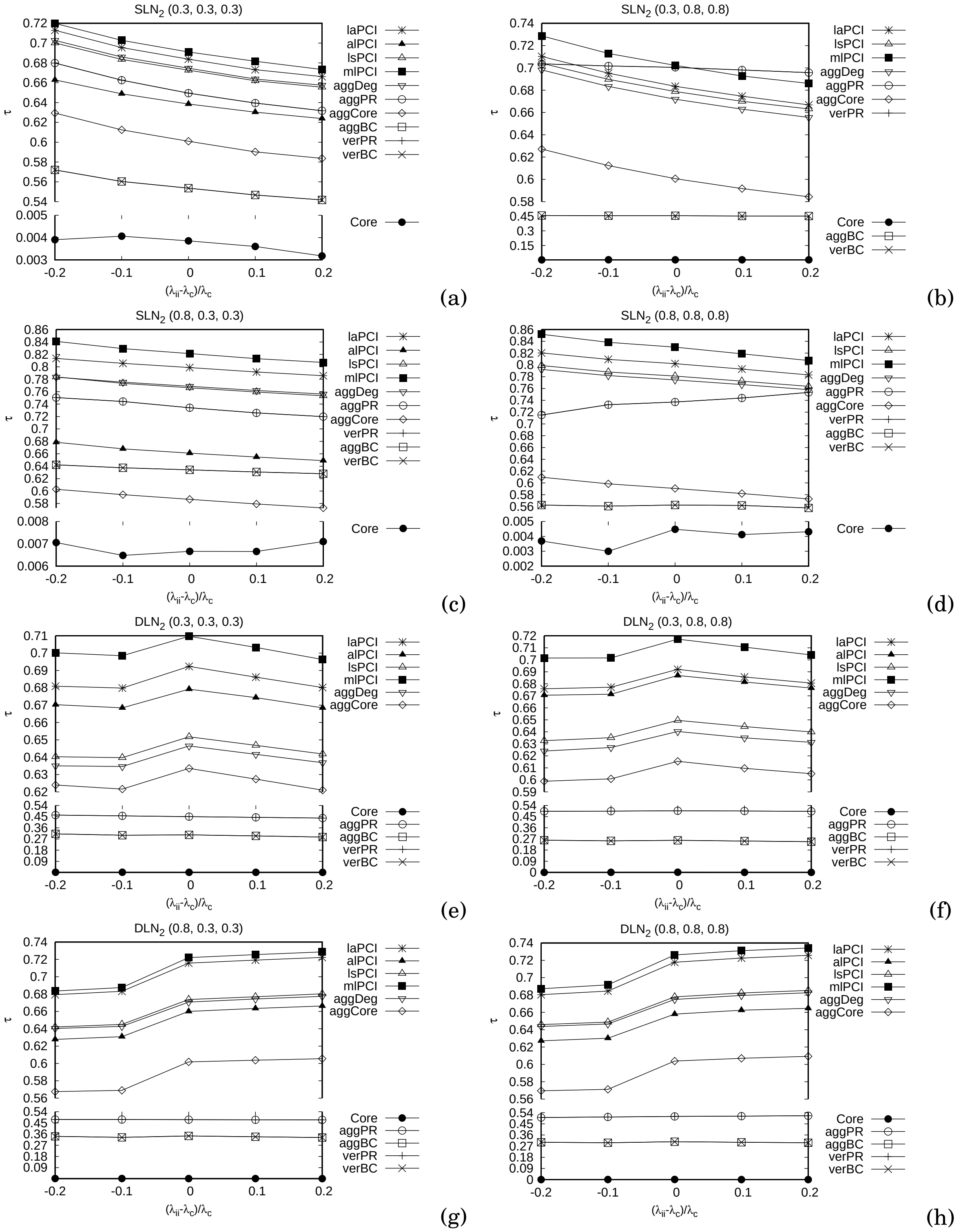
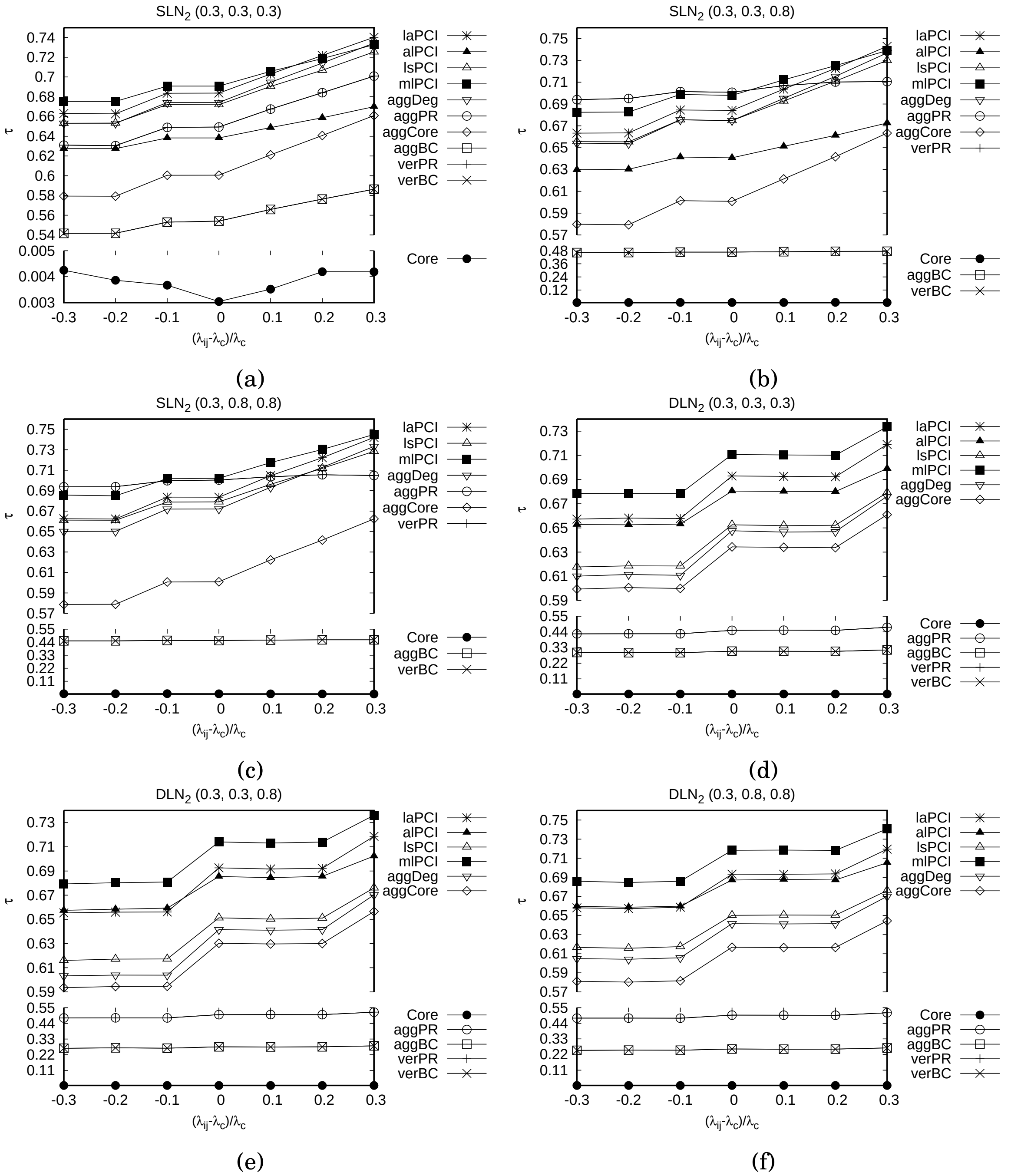


Figure 4.6: Rankings capabilities (Kendall's Tau *b*) of all competing techniques in real networks with synthesized interconnections with respect to uncorrelated with influence in these networks, because it assigns to almost all network nodes the same index value.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4.7: Rankings capabilities (Kendall's Tau *b*) of all competing techniques in real networks with synthesized interconnections with respect to λ_{ij}. *mlPCI* remains at the top of the ranking chain. *verPR*'s performance is better in the SLN networks where interconnections are more dense (when compared to the intra-connections) with respect to the DLN networks, and particularly is at its best when s$_{node}$ or s$_{layer}$ is 0.8. It can be observed that measuring the influence capabilities of a node by counting the number of geodesics that pass through that node (*aggBC*, *verBC*) does not yield competitive results.

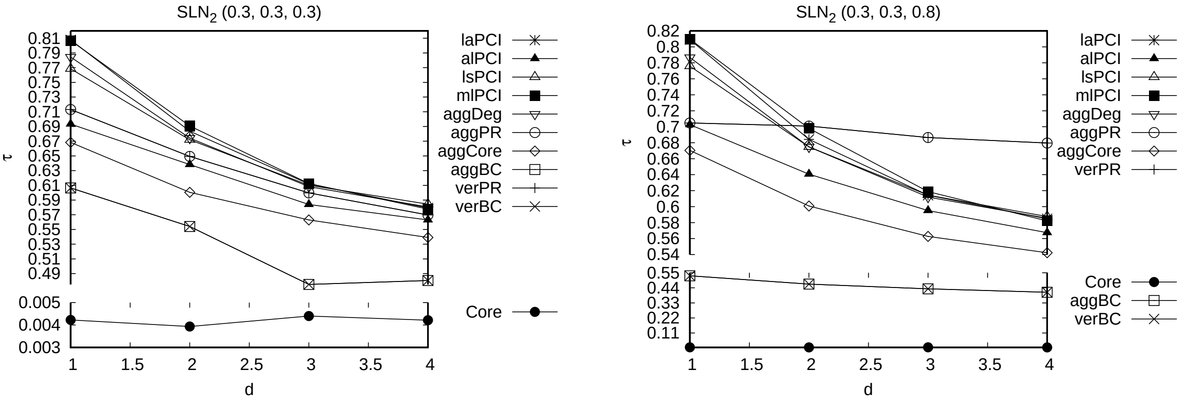


Figure 4.8: Increasing in the number of interconnections in the SLN networks. It can be observed that all methods illustrate a decreasing trend as d increases. Setting s$_{node}$ at 0.8 and thus assigning to a specific set of nodes many interconnections, works in favor of $verPR$ which exhibits an exceptional performance in this case.

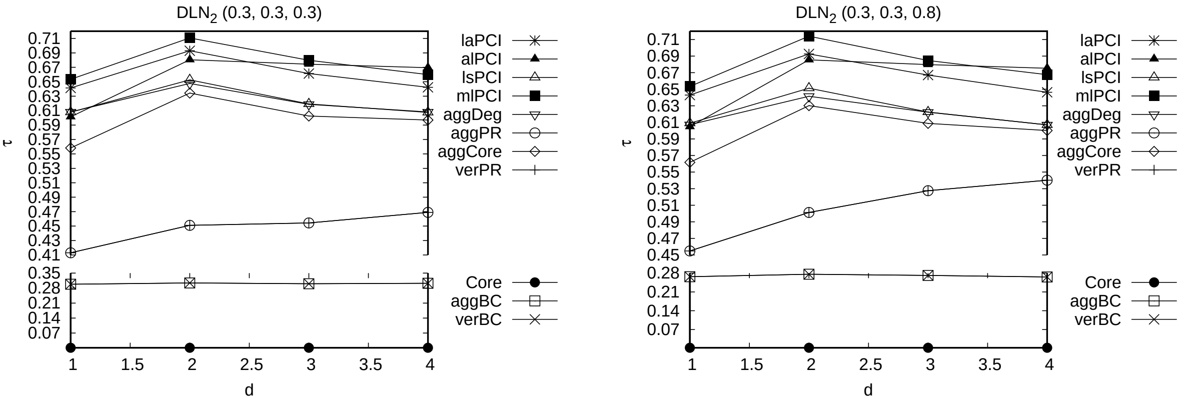


Figure 4.9: Increasing in the number of interconnections in the DLN networks. As interconnections increase $alPCI$ yields better results, i.e., from 4th when $d = 1$ to 1st when $d = 4$. It's performance is different from the SLN networks because for the DLN networks, the distribution of inter-k$_{out}$ is still significantly lower (even for $d = 4$) from that of intra-k$_{out}$ (compare Figures B.1 and B.2 with Figure B.4 in the Appendix) which does not hold for the SLN networks.

Chapter 5

Accelerating Spreading Processes in Vehicular Networks

A Social-based Approach for Message Dissemination in Vehicular Ad Hoc Networks

5.1 Introduction

In this chapter we focus on the selection of relay vehicles−based on tools from graph theory−capable of accelerating the spreading of messages within the vehicular network. One-to-all vehicle communications finds fertile ground in numerous applications in our everyday lives. Consider cases where a driver near a parking lot broadcasts a message regarding limited free spots. Nearby interested drivers may decide to visit this location whereas further away vehicles are less likely to do so. Generally vehicles informed of unfavorable road conditions, for example of blocked roads, traffic jams or accidents, will take prompt action to alternate their route in order to avoid those locations and thus save time and fuel. To this direction the efficient dissemination of messages, that is, the spreading of messages to the largest possible extent within the vehicular network, is of paramount importance.

The main goal of broadcasting in a vehicular network involves the diffusion of messages among the vehicle-nodes, while keeping the number of redundant re-transmissions minimum.

Related publication [**C5**]: Alexandra Stagkopoulou, Pavlos Basaras, Dimitrios Katsaros. *A Social-based Approach for Message Dissemination in Vehicular Ad Hoc Networks*, **Proceedings of the 6th International Conference on Ad Hoc Networks, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (LNICST)**, vol. 140, Springer, pp. 27-38, Rhodes island, Greece, August 18-19, 2014.

This domain has rich literature. Centralized broadcasting, where each node is aware of the entire network topology [182] implies vast communication cost for dynamic networks such as VANETs. Geocasting [80] is another broadcasting approach for the spreading of messages, notification, etc., to wireless nodes located in a specific geographic region. Other studies include the use of connected dominating sets (CDS) [119] to extract a 'backbone' image of a network. Nonetheless, the VANET is a unique network composed of highly mobile nodes with intermittent connections, and thus maintaining an accurate backbone structure is a costly strategy. More sophisticated approaches include those studied in [121] where the vehicular network is divided in groups of neighbors called clusters. For each cluster a leading vehicle, the cluster head (CH), is elected and assigned with specific functionalities, e.g., rebroadcasting. When a vehicle has a message to send, it communicates with his CH, which is then responsible to rebroadcast the message to neighboring CH's and so on, until the entire network is informed.

Simply rebroadcasting (flooding) all messages that a vehicle receives may cause the broadcast storm problem [190]. Other flooding based approaches include probabilistic models where vehicles decide whether or not to rebroadcast a message based on some probability p. However this setup may lead to scenarios with either too few or too many transmissions. The authors in [81] review such methods for large scale routing protocols. Among other studies, VDEB [147] and BPAB [128] are also considered as message forwarding policies in ad hoc networks, however their implementation in the VANET is still incomplete. The optimized link state routing protocol *OLSR* [188] is a proactive methodology and is widely used in mobile and vehicular networks. OLSR employs a specific set of neighboring nodes, called multipoint relays (MPRs), to re-transmit the required messages instead of pure flooding.

In this article we employ social inspired techniques for selecting relay vehicles. Finding appropriate relay nodes in a vehicular network can be cast into the domain of complex networks and the identification of influential spreaders, that is, nodes that can spread information to a large subset of network nodes [139], [77]. These 'super spreaders' are used to accelerate the spreading process and likewise in the vehicular environment can play the role of a relay. Here, we leverage metrics from complex network theory and particularly propose a novel methodology, namely *Probabilistic Control Centrality (pCoCe)*, for selecting efficient relay vehicles. As a competing method we utilize the MPR selection mechanism of OLSR. Our simulations indicate that there are many scenarios where the minimum selected set of relays as identified by OLSR, cannot reach a sufficiently large fraction of the network.

5.2 Control Centrality

In [116] the authors introduce the concept of *Control Centrality* with aim to identify nodes with the ability to 'control' (drive to a specific state) a directed network, based on an initial input and a 'control goal'. To further elaborate we must first note some definitions. A *stem* on a directed graph, is a directed path consisting of n nodes and $n-1$ edges where nodes appear only once, e.g., $i \rightarrow j \rightarrow k \rightarrow l \rightarrow m$. A *cycle* is noted as a *stem* ending on the initial node: $i \rightarrow j \rightarrow k \rightarrow i$. A *stem-cycle disjoint subgraph*, is a subgraph of the directed network where stems and cycles have no nodes in common. Figure 5.1 illustrates one such scenario for a vehicle A. Generally, the control centrality of a node is defined as the largest number of edges among all possible stem-cycle disjoint subgraphs emanating from the node, e.g., 6 for vehicle A of Figure 5.1.

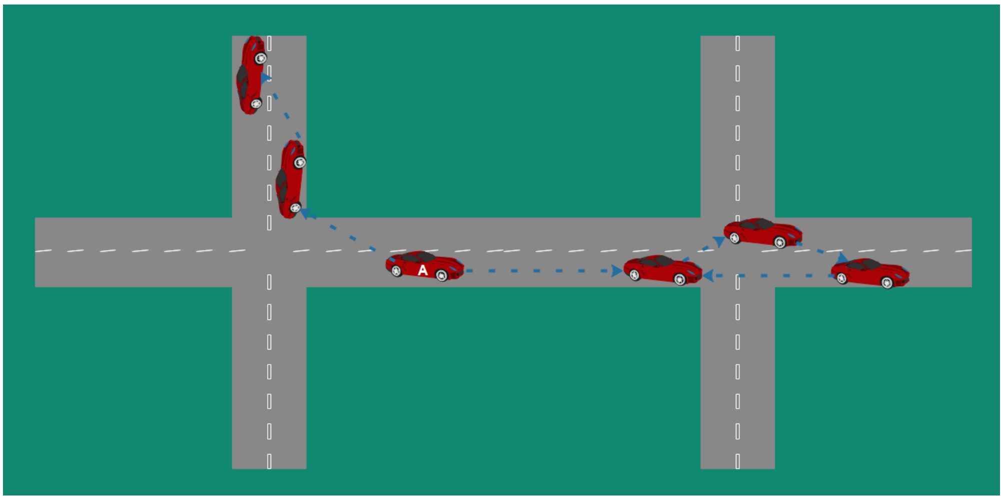


Figure 5.1: Illustration of a stem-cycle disjoint subgraph.

The purpose of this article is to utilize influential vehicle-nodes as multipoint relays. The intuition lies on the idea that those selected relays will rebroadcast a message on behalf of the initial sender and potential inform a large fraction of the vehicular nodes.

5.2.1 From Control Centrality to pCoCe

Initially, we must define incoming and outgoing neighbors in a network of vehicles. Since all connection links among vehicles are considered bidirectional, we use the relative direction between them to classify them as either *in* or *out* neighbors. Generally vehicle A is an out neighbor of vehicle S, when A is moving either in front of S with the same direction or moves away from S towards different directions. Figure 5.2 illustrates the out-neighbors of vehicle S.

Now we can define stems and cycles in VANETs. However, the utilization of cycle paths to enhance a vehicle's importance in a vehicular network is very likely to overestimate its ability in message propagation. To this end, the proposed centrality metric will account only for stems created from vehicle paths. The original algorithm employed stems and cycles that encompass the entire range of a network. However the VANET topology constantly changes (neighboring vehicles increase or decrease their distance, in-neighbors become out and vise versa, etc.) and thus we cannot utilize the method in full range. In this study we confined our work within range of two and three hops (*2pCoCe, 3pCoCe*) from the initial vehicle. Note that *pCoCe* uses

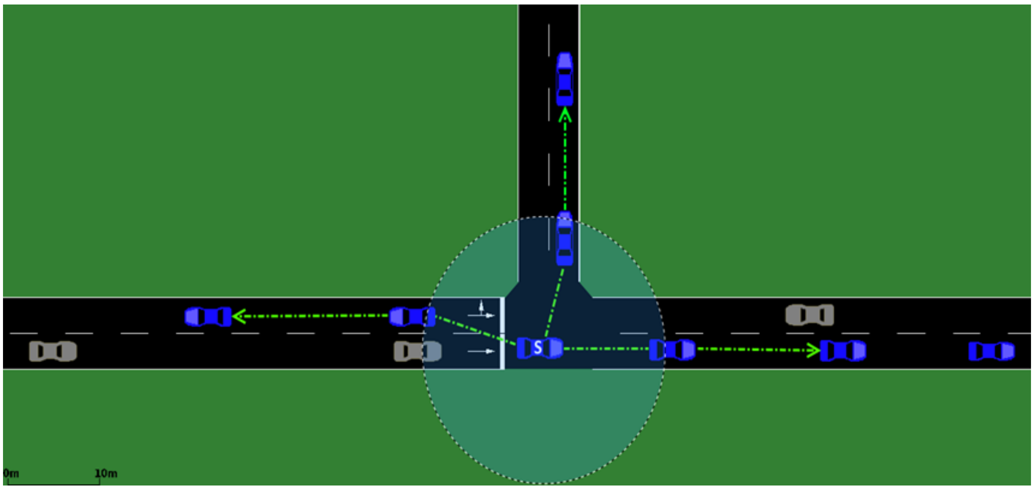


Figure 5.2: The out-neighbors of vehicle S are illustrated.

all stems within our specified range and there are occasions were different stems have common edges. These stems will all contribute in the final *pCoCe* value for a vehicle-node and define its importance in the vehicular network.

The last part of *pCoCe* accounts for the strength of connections between vehicles (*stem power*) and incorporates this attribute in the formed stems. Depending on the quality of the connection for each out-neighbor we assign a weight value between 0 to 1 depicting the strength of connection between the two vehicles. Weights close to 1 depict a perfect communication link whereas values close to 0 depict an almost absent connection, e.g., due to obstacles that interfere with the communication or due to a large distance between the corresponding vehicles. A representative example is illustrated in Figure 5.3.

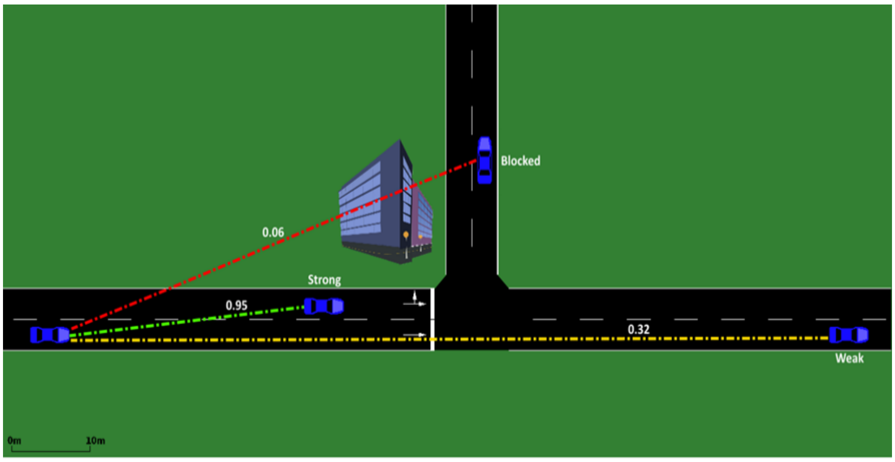


Figure 5.3: Link quality between vehicle nodes.

Finally, the strength of a stem will be computed as follows:

$$Sp = S \cdot PW \tag{5.1}$$

where S depicts the length of a stem in edges and PW is the product of the weights that form it. Further investigation for the strength of connections and its incorporation in Sp is a very interesting task, but it's beyond the scope of this study. Lastly, the *pCoCe* index for a vehicle that will characterize its importance within the network is as follows:

$$pCoCe(x) = \sum_i Sp(i) \tag{5.2}$$

where i denotes the different stems emanating from vehicle x.

In order to accumulate the necessary information for calculating the pCoCe index, vehicles periodically exchange information regarding their relatively close neighbors (and the link quality between them), i.e., their immediate neighbors (2pCoCe) and their next hop neighbors (3pCoCe). This communication ensures that vehicles can build the corresponding paths that will define their significance in the network.

5.3 Relay selection

5.3.1 Selecting relays through pCoCe

pCoCe's algorithm for selecting relays is straightforward. Every vehicle sorts its immediate outgoing neighbors in descending order of their *pCoCe* values, and the neighbor with the maximum value is selected first. In the sequence the next neighbor is examined. If additional two hop neighbors are reached through this new node, the vehicle is included in the relay set. The process is repeated until the entire two hop neighborhood can be reached from the relay nodes.

5.3.2 Selecting relays through OLSR

The same framework for in and out neighbors is also employed for the selection of relays in OLSR. For this technique, vehicles which provide unique access to specific two hop neighbors, i.e., there is no alternative path towards those neighbors, are selected first. Next, the vehicle that communicates with the largest fraction of the remaining two hop neighborhood is selected and so on until all two hop neighbors are reached.

5.4 Performance Evaluation

For out evaluation purposes we employ the vehicular network simulator VEINS [129], which is composed of the traffic simulator SUMO and the network simulator OMNET++.

5.4.1 Simulation design

Grid Network. We evaluated the performance of *pCoCe* in a grid network topology (3X3). Each road segment supports two direction flows and every $2km$ reside intersections with traffic lights. The competitors where evaluated under different scenarios regarding the range of communication, the velocity of vehicles and the density of traffic within the road network. Particularly we experimented with (maximum) vehicle velocities of 14, 20 and $28m/s$ and range of communication at 250 and $500m$. For the density in traffic we introduce a vehicle every 1, 5, 10 and 15 seconds, ranging from very dense to very sparse traffic conditions. The average number of vehicles to the corresponding frequencies is 950, 250, 170 and 120 cars respectively. Vehicles enter the simulation environment from several different road segments.

Communication between vehicles. All vehicles exchange beacon messages every 1 second and become aware of their surrounding cars. A neighbor is deleted from a vehicle's neighboring list if two successive beacon messages are missed. This ensures that each vehicle has a clear and very recent image of its neighboring cars. Additionally all vehicles exchange their neighboring lists, and thus each node is aware of its one hop neighbors and their neighbors and so no, to build 2pCoCe and 3pCoCe respectively.

Spreading process. The evaluation of the competitors is performed upon notification events, i.e., when a diffusion process starts. A notification event is generated from a random vehicle at a random position on the road network (the same vehicle for both approaches) with only one such event existing at a time. The results are averaged over 20 different events for each competing method.

To evaluate the performance of the competing techniques we compute the fraction of the vehicular network (coverage ratio) that received the message-event under different simulation scenarios.

5.5 Results

5.5.1 Experimenting on vehicle density, 2pCoCe

Figure 5.4 evaluates the relay selection methodology of each technique for spreading the message-event at different velocities: 14, 20 and $28m/s$ respectively. The x-axis depicts the density of vehicles in the simulation whereas the y-axis shows the fraction of the informed vehicular network per method. The communication range is set at $500m$.

In the majority of the illustrated results the proposed methodology significantly outperforms the competitor. The fraction of the vehicle nodes "reached" through 2pCoCe are in many scenarios near 80% whereas (on average) OLSR's coverage ratio is below 50%. This is due to the fact that the spreading process as instructed through the relays selected by OLSR "dies out" faster than 2pCoCe's. In the grid network topology the maximum allowed velocity for vehicles does not

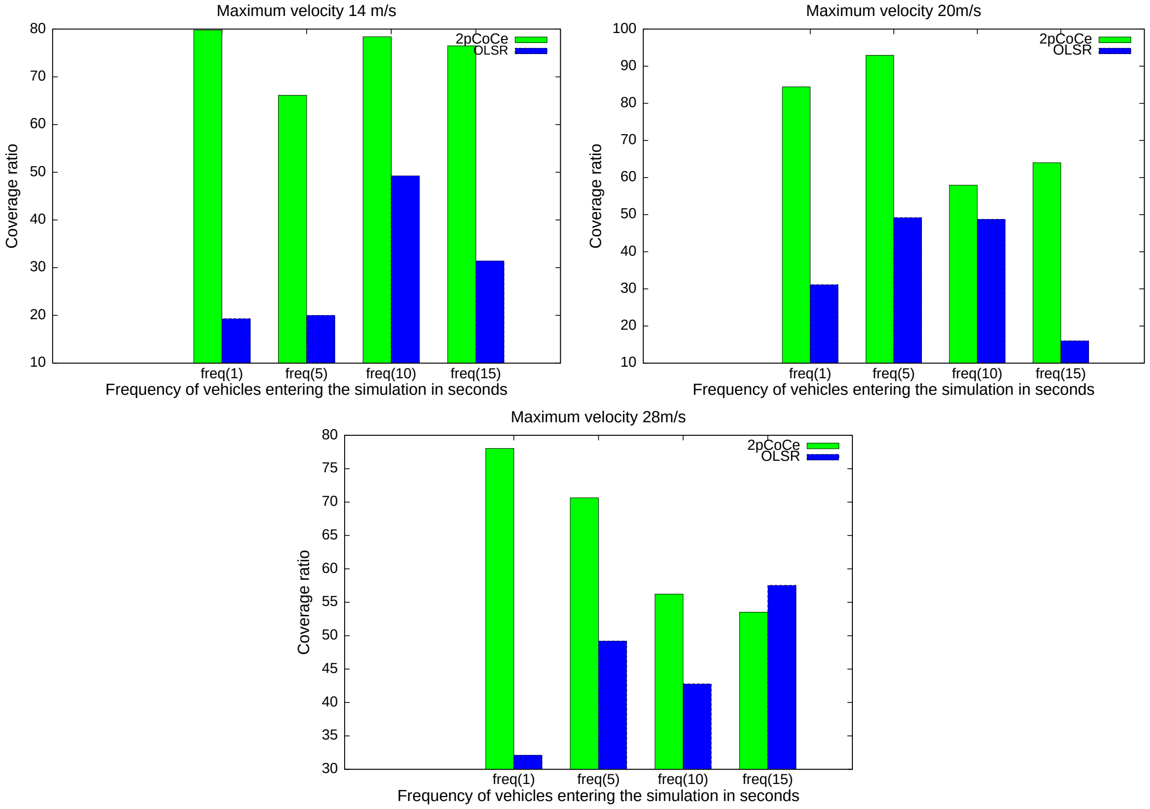


Figure 5.4: OLSR Vs 2pCoCe at different velocities for sparse and dense scenarios.

illustrate a particular trend for the competitors. This phenomenon can be explained by the fact that depending on the scenario, high (or low) speed can have a positive (or negative) effect on the spreading process. For example increased speed can compensate for potentially disconnected parts of a network, while at the same time abrupt changes in velocity can significantly change the immediate vicinity of the vehicle nodes, that is, unexpectedly loose relay spreading paths, and thus limit the outspread of a message.

5.5.2 Differences in the selected relays

In Figure 5.5 we normalize the size of the network that received the message with the number of relay nodes selected by each competing method (y-axis). As already noted, OLSR makes a conservative choice for his MPRs. Therefore, a frequent observed phenomenon is that the spreading “dies” after a few hops (due to false relay set selection) and thus the fraction of the informed vehicle nodes is significantly lower. Since the spreading for *2pCoCe* continues in further broadcasting circles, more vehicles are selected in subsequent steps as relays.

As far as the average number of selected relays per vehicle is concerned, OLSR selects the minimum set of relays. However as shown through our experimentation, in the VANET ecosystem, OLSR results into very poor spreading compared to our approach. For the dense scenarios with

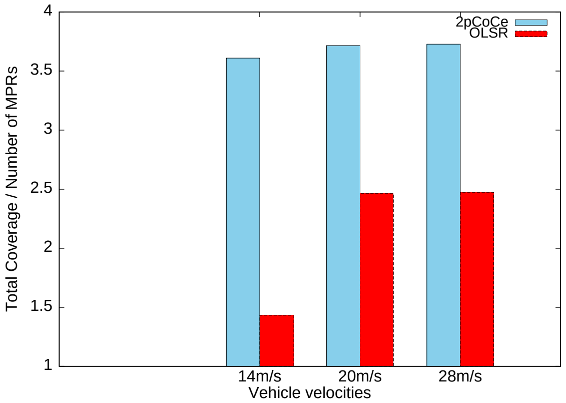


Figure 5.5: Normalizing the coverage ratio of each method with respect to the average number of selected relays.

vehicles entering the simulation every 1 or 5 seconds, *2pCoCe*'s relay set is greater than OLSR's by one or two vehicles whereas for the cases of 10 and 15 seconds we have either equal sets or our set is greater by one. By equal or greater sets we are merely referring to the number of relays selected by each method. Indeed there are occasions were the competitors select similar sets of vehicles, however on average different relays are chosen. Reviewing the differences in coverage rates for both methods in Figure 5.4, one or two additional relays is a good trade-off when a significantly larger part of the network is reached.

5.5.3 Increasing the range of pCoCe to 3 hops distance

In this set of experiments we evaluate the performance of *pCoCe* when increasing the distance of interest from 2 to 3 hops. The results are illustrated in Figure 5.6. When vehicles enter the simulation every 1 seconds, regardless of their velocity, *3pCoCe* influences a larger fraction of the vehicular network. For $28m/s$, the performance of both methods illustrate a decreasing trend as the network becomes more sparse, i.e., when vehicles enter every 15 seconds. Nonetheless, the vehicular network informed by *3pCoCe* is about 63% for the worst case of its performance and up to approximately 73% at best. For this particular case OLSR's performance rises up to about 56%. When the maximum allowed speed is $14m/s$, the performance of 2-3pCoCe seems less affected by the density of vehicles in the road network. Overall, the performance of the competitors is highly dependent on the environs that each relay vehicle faces when the spreading process is active. The proposed method was able to set the right paths for the spreading of messages and inform a significant fraction of the vehicular network, e.g., up to 80%, in many scenarios.

5.5.4 Reducing the range of communication to 250m

Considering only out neighboring nodes for deciding a vehicle's importance (centrality) in a network, can be characterized as a rather unsafe approach. As noted in section 5.2.1 among the

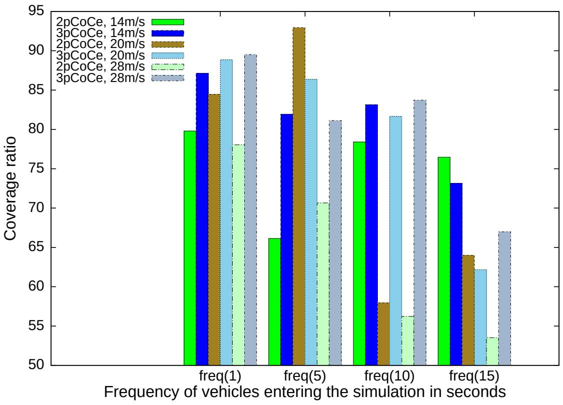


Figure 5.6: Comparing pCoCe's performance with 2 and 3 hops distance.

out going neighbors of a vehicle reside nodes that move away from the sender. Thus these are the vehicles which are most likely to 'exit' the communication range of a sender sooner than other neighbors. Although this phenomenon is highly dependent on their respective velocity and also road topology, reducing the communication range will have a more profound effect for the selection of the relays. In Figure 5.7 we illustrate the obtained results with vehicle frequency set at 1 seconds and communication range at $250m$. Overall, the proposed mechanism outperforms the competitor by informing a larger subset of vehicle nodes. However, at $28m/s$ all methods fail to efficiently spread the message. Analogous results were obtained for 5 seconds frequency whereas for sparser scenarios the performance of all methods was found near 10%. The proposed methodology utilizes vehicle paths of 2 or 3 hops distance for the respective vehicle. These paths are composed of outgoing neighbors and thus further expand the unsafety of out neighbors in additional hops. Therefore, vehicle paths in 2 hop distance should be employed when the communications range is relatively limited, i.e., $2pCoCe$.

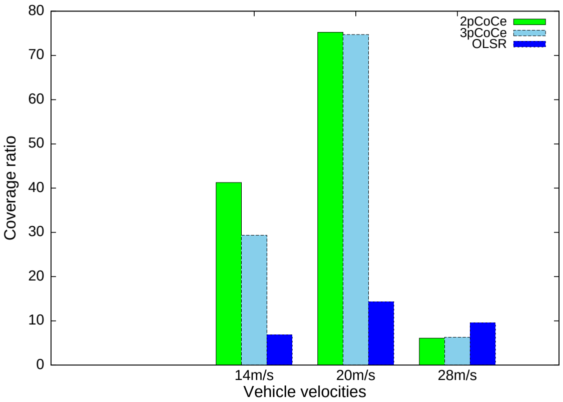


Figure 5.7: Communication range at 250m for frequency of vehicles every 1 seconds.

5.6 Conclusion

In this paper we presented a novel approach for the selection of relay vehicles based on metrics from complex network theory and the identification of influential spreaders. We proposed a novel broadcasting protocol that performs extensively well when dealing with a large number of potential relay choices. Our competitor failed to provide both an adequate coverage rate and reliability as illustrated under diverse simulation parameters. As future work, incorporating the quality of links in the *'stem power'* will provide valuable insights in broadcasting a message under harsh communication environments and different road topologies.

Part III

Blocking the Outspread of Undesired Data in Complex and Vehicular Networks

CHAPTER 6

Blocking the Outspread of Undesired Data in Complex Networks

Dynamically Blocking Contagions in Complex Networks by Cutting Vital Connections

6.1 Introduction

Controlling epidemic outbreaks [169], i.e., the diffusion of "troublesome" contents over the social medium, has received increased attention over the last decade. Most of the so far proposed studies focus on immunization techniques that remove node-users from a network to block the outspread of undesired propagations [88][84][85]. It has been shown that removing the *bridge-nodes* (nodes connected to different communities) or nodes connected to many other nodes (*hubs*), can quite often be an effective solution. However with such methods the immunized entities are completely isolated from the rest of the networked society, while at the same time a network's integrity may be significantly affected. Such drawbacks prompted the research community towards edge-based immunization methodologies for controlling epidemic outbreaks [120][95][156] since the removal of edges is considered as a more realistic approach. For instance removing connections between users, e.g., friendships in Facebook, is a more feasible countermeasure than removing individuals from the entire Facebook society.

Related publication [**C4**]: Pavlos Basaras, Dimitrios Katsaros, Leandros Tassiulas. *Dynamically Blocking Contagions in Complex Networks by Cutting Vital Connections*, **Proceedings of the IEEE International Conference on Communications (IEEE ICC)**, pp. 1170-1175, London, UK, June 8-12, 2015.

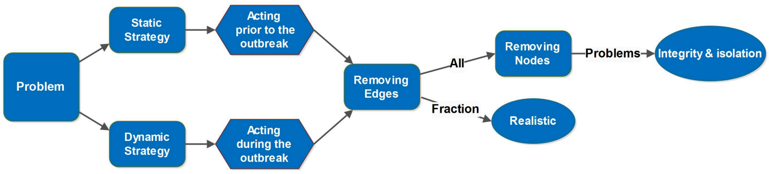


Figure 6.1: Generalized framework for blocking epidemic outbreaks in Complex Networks. This article focuses on dynamic strategies and edge removing mechanisms to hinder the spread of misinformation.

A similar problem to our case study is the issue of identifying a minimal subset of nodes or link connections between them, whose removal will minimize the number of potential infected nodes. Researchers often apply greedy algorithms to address the issue or propose approximations on the basis of greedy strategies [159][86]. While the aforementioned studies focus in deleting network components (e.g., nodes or edges) to protect a networked environment, other studies apply different policies, e.g., by utilizing *protectors* who will disseminate good information to counter a malicious propagation in progress [99].

Removing nodes can be considered as a particular case of edge-based techniques, where the deletion of all connections from a node results in its abscission from the rest of the network. As a next step we group previous works, in terms of how they “protect” a network from a malicious diffusion, i.e., static or dynamic control strategies. A static control approach vaccinates network components prior to the outbreak, by selectively removing a limited β number of nodes or connections, based for instance on different centrality measures or path counting approaches. Although we obtained a number of good strategies for priorly dealing with an epidemic, what more can be achieved by dynamically facing the contagion?

In this article we focus on controlling epidemics by dynamically choosing which connections to remove as we closely follow the contagion within the diffusion steps. At each discrete step a number of β connections may be removed from the network as countermeasures from the authorities. For example consider an event much like KoobFace [146] and a specialized personnel with the knowledge of the currently infected accounts. Instead of taking drastic measures to remove all the connections from the infected users and block the outspread of the virus, the staff could focus on specific interactions among all immediate endangered accounts to hinder or stop the malware from propagating without completely disrupting the networked environment. However we cannot expect for the virus to stay idle while the personnel operates, and thus we assume that we have a limited number of actions (time) before it further propagates.

By mining the knowledge out of a network’s current state, i.e., origin of infection and susceptible surroundings, a more profound and efficient selection among all possible and proximal edges may be adopted, which intuitively will better hinder the contagion. To the best of our knowledge little work is done in confronting an epidemic dynamically. In [72] the authors proposed a dynamic approach for fighting epidemics, but they focused on strategies for healing already infected nodes under the susceptible-infectious-susceptible (SIS) model. Here we follow the contagion as it evolves and propagates through node interactions and propose an algorithm that detects critical connections based on their diffusion capabilities, namely *Critical Edge Detector* (CED). These edges will constitute our targets for immunization in our effort to save the largest possible fraction of a complex network when bounded by a limited number of actions-deletions per step. So far the general framework for blocking contagions is shown in Figure 6.1. Our analysis lies in the lower flow of the diagram.

The chapter sections are organized as follows: in Section 6.2 we provide a formal description of the addressed issue. Next in 6.3 we detail our proposal. Section 6.4 briefly describes the competing techniques and the evaluation criteria as well as the performance of the competing heuristics. Finally in 6.5 the conclusions.

6.2 Problem Formulation

Let $G(V,E,c_e)$ denote an undirected complex network of V nodes connected through E links, where each edge is associated with a positive cost c_e for deletion. The dynamic version of the problem confronts us with the following situation: at each discrete time step t, we have a number of immediate vulnerable nodes which we will try to protect, recovered nodes who were infected in past steps and can no longer be affected by the malicious propagation, and finally the infected ones who will now try to infect their susceptible neighbors. To simulate the diffusion process of undesired data over G, we utilize the susceptible-infectious-recovered model (SIR) which unfolds in discrete steps. Nodes who are infected during the dissemination process are considered as the lost fraction of nodes. Given a budget β of available deletions per step−equal cost for the removal of any edge−we search for those connection whose deletion will result in the least number of lost nodes at the end of the malicious propagation. As a next and final constraint we consider that the authorities exhaust all their available resources at each time step, i.e., resources cannot be saved for later use.

6.3 Critical Edge Detector (CED)

For our method we focus on the infected nodes of each step to create the *Infected-Source-Networks* ($ISNs$), emanating from each individual 'tainted node' x at time step t. The $ISNs$ are created from the susceptible nodes within the n-hop neighborhood of each infected source x (including x) and the link-connections between them, denoted as ISN_x^n. Our work is limited in short distances from the originators in order to fight the contagion near the source of the problem, and inhibit its transition as much as possible. An illustrative example is given in Figure 6.2. Initially we assume that the infection came from nodes $n1$ and $n2$ at time t-1 who successfully infected nodes a and m. The 3-hop infected source networks emanating from the current infected nodes at time t, ISN_a^3 and ISN_m^3, are shown with green and red dashed lines respectively. Note that the infected sources $n1$ and $n2$ from the previous step (t-1) are excluded from our selection in all subsequent steps, since they can no longer contribute in the propagation.

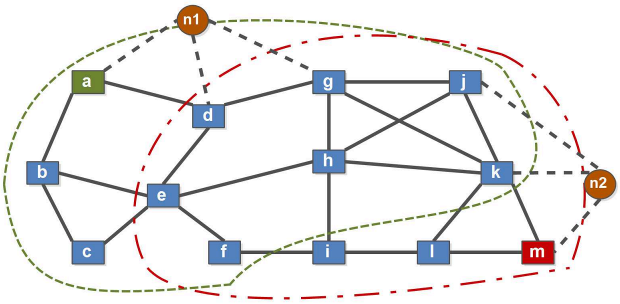


Figure 6.2: In the current time step (t) the infected nodes are assumed to be 'a' and 'm' whereas n1 and n2 are the infected sources of the immediate previous step (t-1) which are now immunized (removed). The dashed lines correspond to the three hop abstract network images, as seen from the perspective of the current infected sources.

To quantify the importance of an edge (i,j) in an ISN_x^n, we calculate the number of shortest paths (using Dijkstra's algorithm) emanating from the infected source x to all other nodes in the current ISN_x^n that (i,j) appears, with respect to the total number of those paths as follows:

$$ISN_x^n(i,j)_t = \frac{sp_{ij}^n(t)}{sp_t^n} \tag{6.1}$$

$sp_{ij}^n(t)$ is the number of shortest paths that the edge (i,j) appears at t step emanating from x, and sp_t^n stands for the total number of those paths.

The concept of *Single Source Shortest Path* is a widely used methodology in network science, well suited for the facet we are addressing in the present study, as we dynamically deal with a contagion directly at its source. At this point we should note that by grounding the source of the infection, i.e., pinpointing the malicious sources, we understand the direction of the propagation. Our proposed technique uses the course-evolution of the diffusion (towards the susceptible environs) to its advantage, and locate those links which will hinder the malicious act to the largest possible extent. However, not all ISNs are of equal size, that is, in the number of susceptible nodes or connections. In fact this is a varying parameter that must be taken into consideration, since edges located in relatively sparse ISNs, may well be overestimated for their spreading potential. Thus we need to include a notion of density for the *end-point* node. Since we noted the course of a virus, the end-point node is a potential direction, e.g., in Figure 6.2, k is the ending node of m-k. The density for the end-point-node j is measured by the formula:

$$d_j = s_j - P_j + \sum_r (s_r - P_r - M_{rj}) \tag{6.2}$$

where s_j is the number of susceptible neighbors of j, P_j corresponds to the fraction of nodes out of s_j with at least one infected neighbor, r depicts the susceptible neighbors of j and M_{rj}

denotes the common neighbors between r and j. If j leads to a dense region of susceptible neighbors, the importance of the connection will be boosted accordingly, whereas for a sparse vicinity d_j will be lower. Finally the final rank for each edge as accumulated by CED is given by the formula:

$$CED(i,j) = ISN_x^n(i,j) \cdot d_j \tag{6.3}$$

Hereafter we assume that the ISNs are obtained from the 2-hop neighborhood of the originator, i.e., ISN_x^2.

6.4 Performance Evaluation

6.4.1 Datasets

A summary for the base attributes of the evaluated networks is listed in Table 6.1. α stands for the epidemic threshold of transmissibility calculated for each respective network [133], and k-core illustrates the largest shells−the core of a network−as identified by the k-shell decomposition algorithm [139]. Various networks were selected for evaluating the performance of the competitors in diverse networked environments; *Hamsterster*: a social network, *Pretty Good Privacy* (PGP): secure information interchange network, *Oregon-2*: an autonomous system graph from May 26 2001, and finally the email contact network, *Enron*. For more details on the evaluated networks please refer to http://konect.uni-koblenz.de/ and [65].

Table 6.1: Network Base Attributes

Network	**No. of Nodes**	**No. of Links**	**k-core**	α(%)	**Type**
Hamsterster	2,426	16,631	24	2.5	Social
PGP	10,681	24,316	31	5.5	Contact
Oregon-2	11,461	32,730	31	5.5	AS
Enron	36,692	367,662	43	1.5	Email

6.4.2 Simulation Design

6.4.2.1 Initiating the Cascade

The origin of the infection, i.e., the initially infected nodes, is an important feature that affects the diffusion dynamics. For instance, if the originators are within a sparsely connected neighborhood, even with a limited number of available deletion per step, the diffusion is very likely to be inhibited. Similar performance will be achieved, if the origin of the infection is placed in the periphery of a network as identified by the k-core algorithm. Such configurations are trivial for our experimentation. On the contrary, if the originators are nodes in denser regions of a network, successfully inhibiting the outspread of undesired data will be more challenging.

To this end, in a similar approach to [95], we initiate the infection from the top-10 most connected nodes (*hubs*) within the highest k-cores of each network. It is safe to assume that initiating the infection from hub-core nodes is no trivial task−maybe the worst case scenario−since the core represents well connected node-users who are “buried” deep within the network structure.

6.4.2.2 Propagation Model

For the diffusion model as noted in [106], the SIS (like flu) suggests no immunity for the interacting nodes, whereas the SIR offers permanent immunization (like mumps). Here we study the penetration of a virus in a networked environment and focus on SIR which unfolds in discrete steps (see Appendix A.1). Briefly, in the initial phase all nodes are in the susceptible state except the initially selected nodes in I. Generally, an infected node at time step t has a single chance to infect its susceptible neighbors and succeeds with probability λ. Immediately after the node enters the R state at t+1 and can no longer be infected in subsequent steps. The process ends when there is no newly infected node, i.e., all nodes are either susceptible or recovered.

6.4.2.3 Removing Connections

In this study we follow the diffusion dynamically, i.e., as it unfolds through node interactions, and thus the links that constitute all possible options for removal at each time step are those in direct contact with the infected sources. As far as the constraint for removing edges per step is concerned, we take 1% of the total connections of each network and name this fraction of edges as *thres*. The x-axis in each plot represents the percent out of *thres* cut in each diffusion step, namely β number of edges. We limit our experiments to small β values per step to evaluate the competitors ability in detecting the most efficient interactions for blocking a malicious diffusion.

6.4.2.4 How to evaluate the performance

In order to obtain unbiased results, for each method we repeated over 1000 diffusion processes. The error-bars in each plot represent the confidence for the interval of the mean, i.e., the true average value is bounded within the specified range. The probability of diffusion among interactions (λ), is chosen based on the epidemic threshold α of each respective network. However in $Hamstester$ due to its lower connectivity we had to use a relatively higher value to obtain significant results.

The impact of each method is measured based on the fraction of the network affected by the false rumor-virus at the end of the SIR process, i.e., number of nodes in R state (lost nodes). The evaluation is carried out in two distinct SIR processes. The first, measures the fraction of lost nodes when no protection algorithm is applied while the second applies the competing techniques respectively.

6.4.3 Competing Methods

The presentation of the addressed issue in this work is original and thus the selection of appropriate competing techniques is crucial. Here we list our selection in the competing methods and also exclude those that cannot be applied. (*i*) Highly connected nodes are noted by many studies as influential spreaders and thus in [156] the strength of a connection is measured by the product of the degrees of its incident nodes ($aDegree$). Note that for this approach only susceptible neighbors frame the degree of a node since these are the nodes we will be trying to protect. For the current competitor the edges are selected in decreasing order of $aDegree$ until β is reached at each time step.

In [95] the authors apply a different approach by strategically selecting which edges to remove, with aim to decrease the probability of a rising cascade. In a similar approach−although under a stochastic and different diffusion model−we apply a strategic deletion of edges to secure the largest number of immediate and endangered individuals.

The first strategy measures the number of infected neighbors each susceptible node has, i.e., it measures how vulnerable a node is to infection in the upcoming step. Nodes with the least number of infected neighbors are firstly treated and so on until the available budget for this step is exhausted. In order to avoid consuming a significant amount of resources to save a single node, for nodes with more than one infected neighbor we remove one connection at a time. If one edge is removed from all vulnerable nodes and there are still available resources we re-initiate the procedure until β is consumed. Note that nodes with only one infected neighbor are completely protected in this round. We name this strategy *alpha* where we try to decrease the probability of a cascade throughout the diffusion steps.

(*iii*) The second strategy, namely $beta$, ranks all susceptible nodes in direct contact with one or more infected sources in decreasing order of their susceptible degree, i.e., number of still unaffected neighbors. With this strategy we try to reduce the number of interaction that lead to highly connected individuals in each step. Note that when the budget β for deleting edges is sufficient to remove the same amount of connections from all immediate susceptible nodes (rare occasion), it applies that $alpha \equiv beta$.

(*iv*) Finally a random selection of edges (*random*) is used as a baseline to create a lower bound of performance. Here a uniform selection among all possible links is applied.

6.4.4 Results

6.4.4.1 Increasing in the number of deletions per step

As a first step to our evaluation we illustrate the results from Figures 6.3 to 6.6. The y-axis represents the fraction of saved nodes, i.e., the percent of node-entities that each respective method managed to secure, with respect to the unblocked outcome of the propagation. It can be

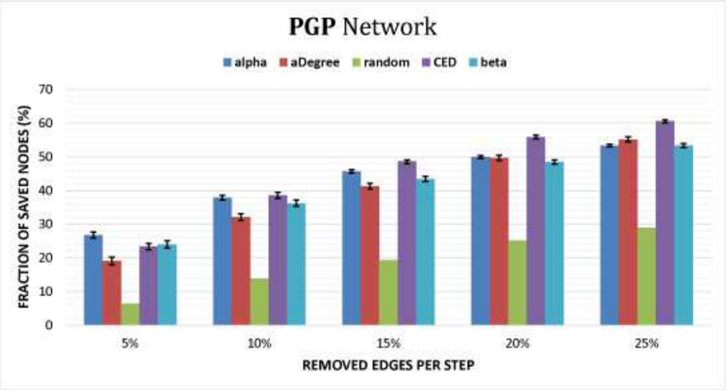


Figure 6.3: The strength of the propagation is 6%. The initially infected set is connected to the immediate vicinity with 548 connections whereas the lost fraction of nodes for the unblocked diffusion is about 280 nodes. As we increase in the x-axis *CED*'s better performance becomes more evident.

seen that the proposed identification technique performs extensively well in most of the observed cases. Our results indicate that cutting of edges within certain limited regions of a network (the *ISNs*) that reside in many shortest paths, is the most effective solution for blocking or hindering the infection dynamically.

To better analyze the performance of the competitors, let us consider the evaluated networks with respect to the connectivity of their initially infected core. The selection of the initial infected seed set out of the most connected nodes within the core shell of *PGP* and *Hamstester*, form a weakly connected set with average degree of 54.8 and 41.1 respectively. It is reasonable to assume for such cases, that by blocking the diffusion directly at its source, a relatively good performance would be achieved by all methods. The results illustrated in Figures 6.3 and 6.4

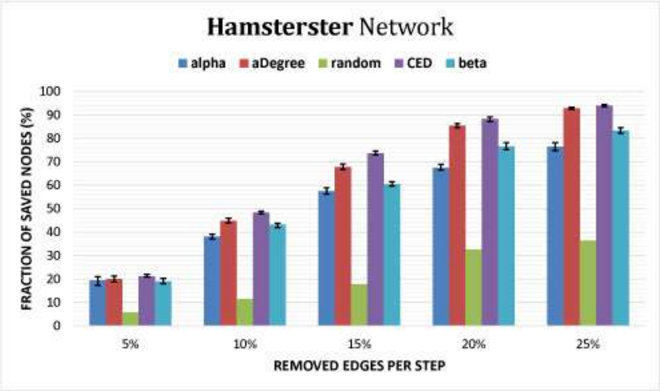


Figure 6.4: The strength of the propagation is 4%. The initially infected set is connected to the immediate vicinity with 410 connections whereas the lost fraction of nodes for the unblocked diffusion is about 360 nodes. For this weakly connected network all methods illustrate a good performance.

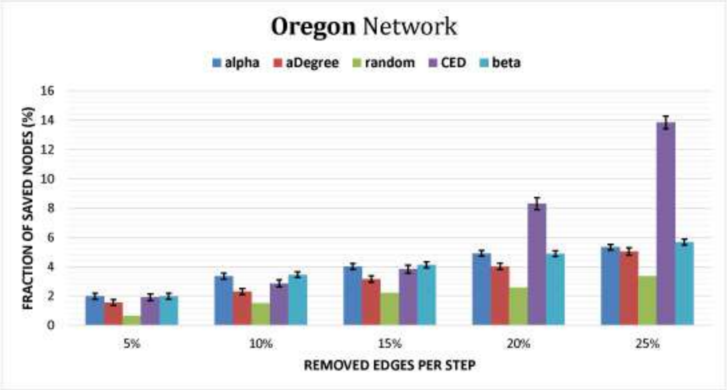


Figure 6.5: The strength of the propagation is 6%. The initially infected set is connected to the immediate vicinity with 3400 connections whereas the lost fraction of nodes for the unblocked diffusion is about 1270 nodes. Only the proposed technique manages to hinder the propagation sufficiently in the later steps of β.

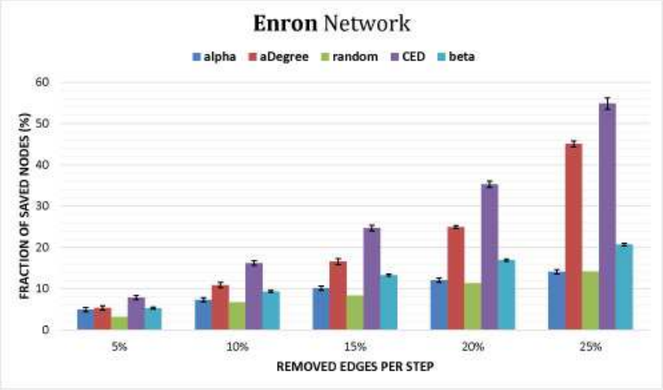


Figure 6.6: The strength of the propagation is 2%. The initially infected set is connected to the immediate vicinity with 11285 connections whereas the lost fraction of nodes for the unblocked diffusion is about 2080 nodes. Again the network is better protected by CED.

confirm this hypothesis. For the PGP network, CED's better performance becomes more evident as we increase in β, whereas in Figure 6.4, $aDegree$ performs equally well with CED.

For the $Enron$ and $Oregon$-2 networks in Figures 6.5 and 6.6, we analyze a more ambitious case, i.e., the average connectivity of the initially infected nodes is 340 and 1128.5 respectively. In these scenarios we expect a more challenging behavior. As illustrated, the fraction of saved individuals is significantly less from the previous network cases. For the lower values of β: 5,10 and 15% it appears that none of the evaluated techniques is able to block the contamination significantly, i.e., the saved individuals are less than 5% in $Oregon$-2. Only CED manages to save up to about 14% from the fraction of lost nodes when β=25%, while the rest of the evaluated techniques illustrate similar behavior near 6%.

Similar results are also reported for the $Enron$ network. The virulence of the propagation is set at 2%. For both $alpha$ and $beta$, we observe little improvement in the saved individuals as we increase in β, while CED, as usual, outperforms all competing techniques. To our interpretation, although both strategies performed relatively well for the rest of the experimented networks, it seems that trying to reduce the probability of a cascade by decreasing the overall connectivity that lead to infected sources or targeting those links that lead to the most susceptible nodes, is not efficient when applied in the core of a well connected network as in this particular case.

Overall we attribute CED's better performance to the following remarks. First, although there are occasions where the contagion cannot be completely stopped in the early steps (due to the infection being rooted deep within a well connected network), by removing the edges as identified by our approach we force the malicious propagation towards longer interacting paths. Thereby more resources can be used in the next steps to inhibit its transition and stop its outspread to more distant regions of a network. Second, by measuring the density of the surroundings of the end-point node, we alleviate traditional drawbacks of shortest path algorithms, since our method will discount the significance of otherwise important links which lead to sparsely connected parts of a network.

6.4.4.2 Increasing in the virulence of the malicious propagation

For our final evaluation in Figures 6.7 and 6.8, we investigate on the performance of the competing techniques, as we increase in the strength of the malicious propagation, i.e., increase in λ. As usual, λ is selected near the epidemic threshold of each respective network [77][139]. We focus on the results of PGP and $Oregon$-2, that is, one network of each category with respect to the average connectivity of their initially infected set from the core. Similar qualitative conclusion were obtained from the remaining networks as well. β is set to its largest value, i.e., 25%.

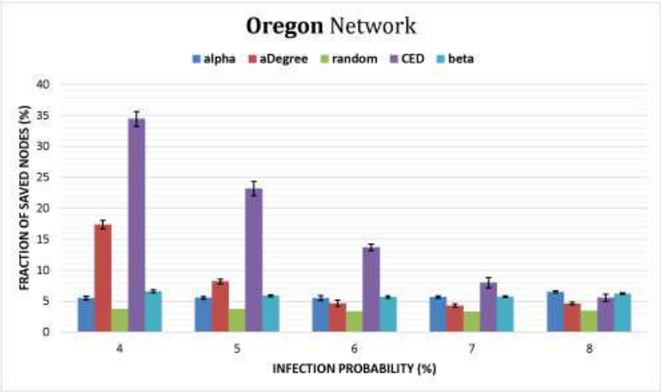


Figure 6.7: The y-axis represents the fraction of saved nodes with regard to the lost nodes of the unblocked diffusion (814, 1048, 1270, 1488, 1714) respectively. Our approach seems to be affected by the increase of λ significantly later than its competitors.

In order to measure the influence capability of nodes in complex networks, a problem formally known as detecting influential spreaders in complex network structures [77][58], the virulence

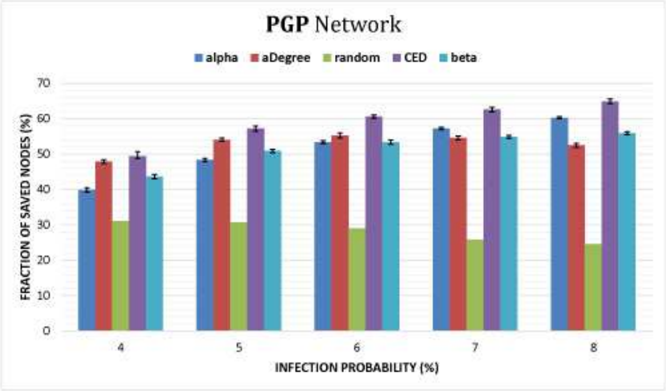


Figure 6.8: The y-axis represents the fraction of saved nodes with regard to the lost nodes of the unblocked diffusion (113, 190, 280, 385, 511) respectively. CED illustrates better results by securing a significantly larger part of the network's interacting nodes for all λ values.

of the diffusion should be kept in relatively low values. This is due to the fact that for larger infection values, an epidemic occurs regardless of the characteristics of the node elected as the origin of the infection. In this study, where we initiate the infection from multiple sources from the most connected nodes of the core of each respective network, we expect that blocking the malicious propagation as λ increases will become a very challenging task.

The results in Figure 6.8 indicate that when the network is sparsely connected, the infection can still be significantly mitigated, even when the virulence of the diffusion increases above the epidemic probability. For the lower λ values $aDegree$ and CED illustrate similar performance, however as we increase in λ the proposed technique significantly outperforms all competing methods. $aDegree$ is affected by the increase of λ around 6% and henceforth its performance starts to decent, whereas reducing the probability of the cascade with both $alpha$ and $beta$ strategies, seems to have an increasing performance that surpasses $aDegree$ when above the epidemic threshold. Nonetheless further increasing in λ will only decrease the fraction of saved nodes that each respective method manages to secure.

By following the performance of the competitors in $Oregon$-2 we observe a different outcome. For this scenario all methods illustrate a decreasing performance as λ increases. However the competitors fail to protect an adequate fraction of the network nodes even bellow the epidemic threshold. Only the proposed technique bears more resistance to the virulence of the propagation and is able to save a significantly larger number of endangered nodes. To our understanding when λ increases beyond a certain threshold−different for each network depending on its connectivity−the diffusion cannot be significantly hindered. This is due to the fact that even by deleting a large number of immediate connections, i.e., increase in β, and thus significantly diminish the available paths from infected nodes to susceptible individuals, when we are bound to the higher values of λ the virus is very likely to survive even within the now few remaining interaction.

6.5 Conclusion

In this study we take a first step in confronting the diffusion of malicious contents over networked populations dynamically, while we follow the virus as it progresses through node communications. Most of the so far proposed techniques focus on static strategies, however we believe that the problem is dynamic in nature and must be addressed appropriately. We proposed an algorithm that utilizes well studied heuristics from the literature of graphs, which was found to be quite effective in blocking the outspread of the diffusion. We used a number of representative competitive methods and strategies–what we believe baseline approaches for the dynamic facet of the addressed problem–to evaluate the impact of our method. Our technique was found to be more efficient by securing the largest fraction of individuals almost in all observed scenarios. Finally we conclude that when increasing the strength of the prorogation above the epidemic threshold, successfully hindering the propagation can be a very challenging task in a well connected network. Nonetheless *CED* illustated a more resistant behavior in the increase of the virulence of the propagation.

Chapter 7

Blocking the Outspread of Undesired Data in Vehicular Networks

Blocking Epidemic Propagation in Vehicular Networks

7.1 Introduction

In this chapter we further discuss on the vehicular network and potential emerging threats shading the vision of the always connected car. Up to this point we have elaborated on the tremendous benefits brought to our everyday lives from the prospect of communicating vehicles. Nevertheless, having the cars connected over an ad hoc network does not come free of dangers; a compromisation of the car's security/defense system can give control to third parties over it. This is a feature that any 'computerized' car can suffer. Carjacking [43] events gradually appear in the news [1] and technical magazines [34]. While these incidents are currently limited, the availability in the near future of millions of vehicles with V2V capability raises the danger of 'epidemic' outbreaks over VANETs, where malicious software will infect large number of cars invalidating the benefits of V2V technology and even causing human casualties.

The study of epidemics has a long history in medicine and related areas [192], and has recenty seen a tremendous flourishing in the computer science realm [42] due to the great expansion of wired/wireless networks and portable devices and also due to the widespread use of online social networks (e.g., Facebook).

Related publication [**C2**]: Pavlos Basaras, Ioannis-Prodromos Belikaidis, Leandros Maglaras, Dimitrios Katsaros. *Blocking Epidemics Propagation in Vehicular Networks*, **Proceedings of the 12th IEEE/IFIP Annual Conference on Wireless On-demand Network Systems and Services (WONS)**, pp. 65-72, Cortina d'Ampezzo, Italy, January 20-22, 2016.

Among the issues pertaining to epidemics in computer networks, the topic of blocking the expansion of an epidemic has received significant attention reflecting the importance of protecting the unhindered operation of networks. However, the study of epidemic outbreak control so far has focused on: a) centralized methodologies where a network controller can make decisions over the network topology, b) on static or semi-static networks with no or very limited node mobility, and c) on the feasibility of the node or link removal operation which can take a node out of the network [27]. As far as existing VANET research on this topic is concerned, this has almost exclusively focused on modeling of the worm spreading process under various traffic conditions [101], [135], [173] and a scheme for patching the infected vehicles using cellular network's connectivity [101].

7.1.1 Motivation and contributions

Unfortunately, the aforementioned assumptions made by the existing works on epidemic control have little or no applicability at all in the VANET environment. A VANET is a highly distributed environment with opportunistic communication among vehicles, and clearly a fixed/centralized element (e.g., road-side unit) can not easily – due to cost and installation constraints – play the role of a detector and/or disinfector; even if a cellular network is provided for delivering patches, the density of infected vehicles in a region may prove to be a challenging environment for the base station to detect the malicious software and/or remove it. Moreover, the volatility of the network topology due to high vehicle mobility creates opportunities for effective blocking of the expansion (in case the infected vehicles are within an isolated component of a partitioned network) or make the blocking of it an extremely difficult task (in case that many infected vehicles are quickly moving across all 'parts' of the network). Finally, it is not clear how could an infected vehicle be "thrown out" of the network, as it is done in static computer networks where part or all of the communication links of the infected computer are cut down or as it is done in human populations, where an infected individual may be quarantined; in VANETs, an infected vehicle may/can continue to transmit even if it is infected, continuing to spread the infection.

This article adopts a different perspective in the study of epidemics in the VANET environment by separating the task of infection blocking from the task of disinfection. The latter is highly dependent on the kind of software that creates the infection, on the particular type of vehicle that needs to be disinfected, and on the existence, coverage and capacity of wireless networks in the area of infection spreading; for instance the infected vehicle may need to be taken to a specialized car service point to be disinfected. On the contrary, the former task can be performed in-situ in a distributed fashion with the cooperation of other vehicles and minimal use of fixed infrastructure, and most importantly, techniques developed in the discipline of network science can be used for limiting the spreading of the epidemics. The present article proposes a cooperative technique which is the first one in the literature that utilizes V2V communications to "black-list" some (or potentially) infectious vehicles, and thus refrain other vehicles from accepting for processing packets transmitted by these vehicles. This technique can be seen as a node/link removal algorithm for blocking contagions appropriate for vehicular environments.

The present article makes the following contributions:

- It introduces the problem of blocking contagions in vehicular environments under the new perspective of separating the epidemic's blocking from the curing process.
- It introduces an epidemic blocking technique which is (almost) fully distributed making minimal use of fixed infrastructure to combat the expansion of the malicious software.
- It evaluates the proposed technique via simulations using established simulators to study its efficiency across a range of values of the most significant independent parameters that impact the performance of the method.

7.2 Related work

The present work is of relevance mainly to the topic of malware epidemics in VANETs and in complex networks in general, less related to the topic of security threats in VANETs, and remotely related to the defense methods for reliable vehicular communications. Worms can easily propagate through a network without any human intervention, and in recent years they have emerged as one of the most prominent threats to the security of computer networks [39], [178]. Effects of worm epidemics on VANETs have been recently studied in [101], [135], [180] and the common conclusion is that they pose a high level of danger; a worm attack on a VANET may interfere with critical applications such as engine control [140] and safety warning systems [43], hence resulting in serious congestion on the road networks and large-scale accidents.

There is an extensive body of literature on combating infections' expansion in complex networks based on node-removal methods [35], [88], based on link-removal methods [27], [95], [159]. Nevertheless, these works are not directly applicable in vehicular environemnts for the reasons explained in subsection 7.1.1 or because the proposed countermeasures [84], [163] do not fit a VANET.

In the area of security threats for VANETs, there are numerous kinds of attacks that may affect the reliable communication among the entities of a VANET such as Denial-of-Service (DoS) attack, fabrication attack, alteration attack, replay attack, message suppression attack, sybil attack [185]. Except from different kinds of attacks in terms of the used mechanism, there exist also other categories. For instance, a selfish driver could try to take advantage of the received information for personal benefit, while on the other hand a malicious attacker [164] aims to harm the users or the network with no profound personal gain.

A substantial amount of research on defense mechanisms has focused on intrusion detection systems for early detection of malicious nodes [37], [143], [160]. Regarding which, both specification-based [160] and anomaly-based treatments [143] have been investigated. Moreover, an attempt to deflect attacks using honeypots has been described in [162]. Finally, new techniques for filtering out tweaked data have been recently developed [28].

7.3 Virus Propagation

The spreading process in complex networks is a widely studied topic that finds applications in varying disciplines [91]. Of particular importance is the problem of information propagation over complex networks, e.g., how information ”travels” over networked populations such as Facebook. A well established and widely used model describing such processes is the *Susceptible-Infectious-Recovered* (SIR) model (see Appendix A.1). SIR is employed for simulating the propagation of a virus in the vehicular network. Particularly, a vehicle that can be affected by a virus will be a susceptible (**S**) vehicle. Infectious (**I**) vehicles will try to infected their current neighborhood, whereas recovered (**R**) ones, are either vehicles that cannot get infected (cf. 7.5.4.5) or those that have received a ”cure”, i.e., a patch that removes the virus and immunizes the vehicle in further contacts [101]. Unlike static networks, VANETs are characterized by a constantly changing topology due to transmission range limitations, obstacles or limited by geographic proximity and road topology. A vehicle becomes aware of its current neighboring vehicles through frequent exchange of beacon (*heartbeat*) messages and thus, the target set of an infectious source changes over time; from sparse to dense neighborhoods and vice versa which evidently affects the diffusion dynamics.

In wireless networks nodes can communicate in a one-to-one fashion, i.e., *unicast*, one-to-some, i.e., *multicast* or one-to-all, i.e., *broadcast* communication. In a similar way we assume that a potential threat will follow one of the above mentioned methodologies to propagate to the next target(s). In our framework we focus on broadcast propagation. Finally one last characteristic that needs to be taken into consideration is the number of contacts, i.e., transmissions between I and S vehicles, necessary for the virus to propagate. This final attribute will stand as a virus specific parameter regarding the strength of the virus, e.g., the length of the worm code or the way it is hidden within the exchanged messages. Hereafter we will refer to this attribute as the infection delay (τ) [101].

7.4 Proposed Mechanism

7.4.1 Specialized Hardware (SH)

In this work we separate the functions of disinfecting from detecting infectious vehicles, and focus on the later. Our approach requires a specialized hardware, namely SH, which will play the role of the detector and identify infected vehicles within its scanning range. We envision the SHs as stationary scanners and coordinators between the communicating vehicles rather than entrusting cars with that functionality. This is due to the fact that exploited security flaws that are severe enough to require physical interference to get rid of the infection, in occasions much like [2], can be more efficiently handled in a stationary SH. Thus, we conceive the SHs as highly secure devices initially deployed in a similar manner to Road Side Units (RSUs), that communicate and scan vehicles over the wireless medium.

In a wireless network when you have to keep the transmission power within acceptable limits, the overall efficiency of the network can be improved by either reducing the transmission rate or reducing the transmission range [170]. Based on this basic rule of thumb, and since the SHs must exchange high volumes of data with the vehicles that are under inspection, the transmission range of the SHs is kept low in order to be able to achieve high throughput. Only that way we can reassure that the vehicle can be fully scanned and correctly identified in terms of infection during its contact time with the SH.

7.4.2 Isolating Infectious Vehicles

Based on the fact that we can only detect malicious nodes as long as they are in the vicinity of an SH, it is not straightforward that the whole vehicular network can be protected. In the current work we assume that the SHs are only capable of identifying infected vehicles, but they are by no means capable of revoking the license of cars to participate in communication protocols [167]. Moreover, we expect that potential viruses attempting to spread over the network will be newfangled, i.e., there are no ”predefined medicines” and thus a questionable amount of time may pass until an appropriate patch is ready for dispatching. Nonetheless, even if vehicles have some sort of access to a cellular service (e.g. $4G$ communication) enabling them to download and install a patch in sort time, there may be occasions where physical access to the car is necessary in order to carry out the hack, e.g., the Tesla case [2].

Our primary concern is to effectively mitigate the spreading of a virus in a vehicular network, until an appropriate patch arrives or “physical” treatment is administered. Although we may not be able to heal a vehicle, we are capable of informing the rest of the vehicular network for its presence. Thus, SHs are also responsible for broadcasting the list of the so far identified infected vehicle ids, i.e., a *Black List* (BL). Hence, each healthy vehicle that ”hears” the BL is instructed to shut all communication with those vehicles.

So far several considerations emerge. First, a vehicle that has not yet been in contact with an SH has no knowledge of the infected ids, and thus still stands unprotected against an (already identified) infected neighbor. Moreover, in each vehicle different versions of the BL may exist, depending on their last contact with an SH and the potentially newly identified infected vehicles in that interval. Thus the problem of outdated BLs arises. To this end vehicles are instructed to exchange their versions of the BL list, compare their own version with that of their neighbors, and hence cumulatively increase the awareness of their own and their near vicinity for the infected sources. This extension has a twofold benefit; first, isolated areas, i.e., areas relatively far from any SH, may yet be protected if an informed vehicle traverses the area. Since we will be able to deploy a limited number of SHs (due to infrastructure costs), vehicles must fill such ”void spaces” by circulating the list. Second, the BL version of each vehicle is no longer based on the timestamp of its last contact with an SH, but is swiftly updated to the BL of the neighboring node with the most recent timestamp. Thus the possibility of significantly outdated BLs is minimized. Figure 7.1 is a simple illustration, where an infected vehicle A enters the range of the SH and infection is detected. Upon detection the SH broadcasts the list of all infected vehicles−currently only vehicle A−which is heard from vehicle B and so on.

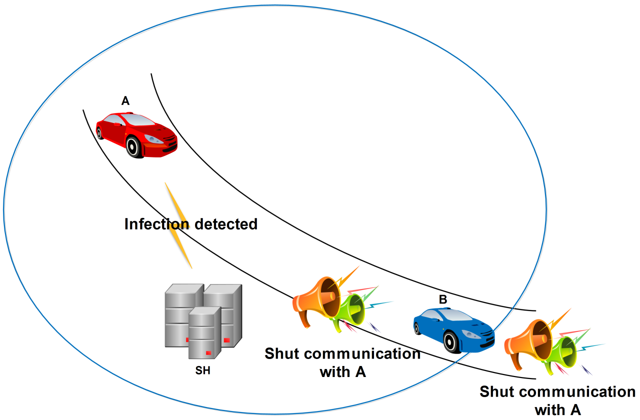


Figure 7.1: Vehicle B is informed of A's infection by the SH. B will further broadcast (and exchange) its version of the BL with all other vehicles found in its trajectory.

Up to this point we detect and inform the vehicular network for the presence of infected vehicles, or in other words we *remove* nodes from the vehicular network. In correspondence, blocking epidemics in complex networks is a broadly addressed problem, where−among other techniques−researchers remove important nodes based on centrality measures, e.g., the degree centrality, to block the outspread of undesired propagations. It was found that removing the most connected nodes, the *hub* nodes, is a low cost and quite effective method. We cannot find complete equivalence in the different frameworks due to the very nature of $VANET$s, we can however exploit several points.

In [27] we proposed a method for blocking epidemics dynamically, i.e., during (and not prior to) the outbreak. Similarly, upon detecting an infectious vehicle, the BL is updated and circulates within the network. So far through the proposed mechanism we diminish further damage that infected vehicles would exert in the system, if left undisturbed. Unfortunately we cannot estimate the time of infection of the identified vehicle, i.e., was the vehicle infected just a while ago or long before? In either case there is strong possibility that nearby vehicles (yet not scanned) are also infected. Hence maybe we can further protect the network by being cautious against the infected node's vicinity. To this end we maintain a second list, namely *Potentially Infected Vehicles* (PIV), where we include either all or a $fraction$ of an infected vehicle's current neighbors. Hereafter, we will refer to vehicles in BL as β, and respectively as π to those in the PIV list.

Similarly to the BL, PIV will be broadcasted from both vehicles and SHs. The difference between the two lists, is that vehicle ids in PIV are only temporarily banned from the system until those vehicles are scanned. Hence, once a π vehicle enters the range of an SH we have two possible outcomes. If the vehicle is found ”clean”, it is simply removed from PIV and its communication is restored. However if infection is detected, the vehicle is converted to β type (moved to BL) and all of its neighbors become π vehicles.

When the entire one hop neighborhood of a β is added in PIV the procedure is straightforward. However when only a fraction of those nodes is included, certain decision rules must be chosen that meet two basic criteria; fairness and efficiency. First, as we discussed earlier, removing highly connected nodes can be quite efficient in blocking the outspread of undesired propagations, or in other words those nodes can be very effective spreaders. Hence, choosing neighbors in decreasing order of their degree until the ”cut”, i.e., the desired fraction of neighborhood is attained and included in PIV, is our first intuition. Second, vehicles who had been in contact with a β car for a longer period, have a higher probability to be infected than more recent neighbors, especially for cases of large values of τ. Hence nodes are included in PIV in decreasing order of their contact duration, i.e., the oldest neighbor is included first and so forth.

A more sophisticated approach accounting for infected vehicles which meddle with the defense mechanism, i.e., meddle with either the BL or PIV or both, by broadcasting empty lists or meddle with the ids within, is beyond the purpose of the current study and is left for future work. In this article, we try to protect the vehicular network from a potential virus spreading though vehicle nodes, by initiating another spreading process to counter its effect. This facet is formally known as *competing memes propagation on networks* [103], where the meme, i.e., the virus or the list, which reaches/influences more nodes wins. Our intuitions lies in the belief that if we can inform a large number of nodes−through SH and vehicle (re)broadcasts−for infected and potentially infected nodes, we can significantly mitigate the spread of a worm-virus.

7.5 Experimental Design

7.5.1 Simulators

For the evaluation of our model, we use the simulator VEINS [129], which is composed of two well established and widely used simulators; OMNET++ an event-based network simulator and SUMO, a road traffic simulator.

7.5.2 Map

Integrated within VEINS, is the map of a city in Germany, namely *Erlangen*, which we used for our simulation. Figure 7.2 illustrates our experimented road topology. It is a rich road network environment of many intersections and different paths leading to various destinations. Note that the red boxes are buildings, i.e., obstacles interfering with the communication of vehicles. The

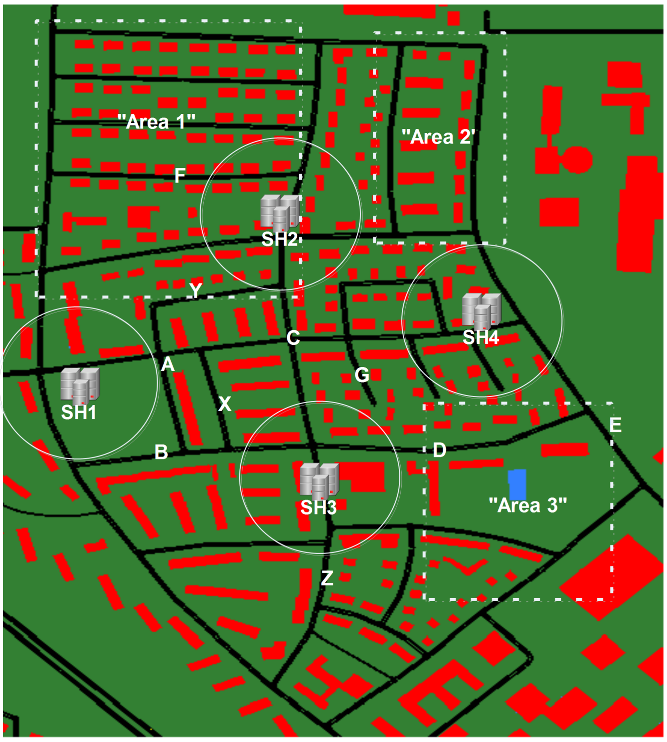


Figure 7.2: Part of the Erlangen city. SHs are positioned near the center of the map. The illustrated scanning region is indicative, to highlight the relatively short range of the specialized hardware devices.

locations of the SHs are also illustrated, however the optimal positioning for a set of n such computing devises is an open issue of many parameters. Setting aside budget constraints, i.e., number of available SH placements, we name just a few variables that we believe should be taken into consideration for an effective placement:

- the popularity of the road segments near an SH, i.e., frequently traversed road segments, namely *density driven* placement
- the number of routes passing through an area controlled by an SH, e.g., shortest paths, namely *topology driven* placement
- or social attributes such as city attractions, i.e., *social driven* placement

Nonetheless, investigating all such parameters individually (or in a combined scheme), is beyond the scope of the current study. In the current framework, we apply a simple allocation for the positions of the SHs by simply focusing in the center of the experimentation environment as illustrated in Figure 7.2. Note that buildings will interfere in both the transmission range of vehicles and the scanning process of the SHs.

7.5.3 Initially Infected Vehicles

As illustrated in [101], a single vehicle is enough to contaminate the entire network. Following the same policy, we initiate the malware propagation from a single spreader. However, our experimentation showed that initiating the infection from different positions yields different results. This is due to the fact that the different vehicles will experience different conditions, i.e., different number of neighboring vehicles, different speeds and directions between them, etc. Furthermore the relative position of the initial spreader and the relative position of the SHs also plays a crucial role for the spreading dynamics of the virus. For instance, if the initial spreader falls within the range of an SH in short time after it starts its malicious behavior, the spreading process is very likely to stop very quickly, especially for the larger values of τ. In our experimentation we avoid such cases.

With the above consideration we experimented with a wealth of different positions for the initially infected vehicle as illustrated with the different points in Figure 7.2, e.g., A, X, etc. The infection starts after running the simulation for 100 seconds whereas the total simulation time is 500 seconds. For each point the results were averaged over 20 distinct runs.

7.5.4 Vehicle Settings

7.5.4.1 Communication

We assume that all cars are capable of communicating with $DSRC$; according to [131] an acceptable communication range for vehicle applications is about $300m$ and this is used in our simulation. This range that can be achieved by low transmission power is enough for the correct dissemination of a message in a neighborhood while it improves spatial reuse in heavy traffic. In rural environments, in scenarios with low data rate (3Mbps) authors in [131] showed that Packet Delivery Ratio (PDR) of 60% can be achieved for such medium distances.

7.5.4.2 Routes & Density

For selecting the trajectories that vehicles will follow in our simulation, we applied the predefined tools within the road traffic simulator SUMO to obtain a diverse range of routes. Specifically a total of 30 different routes were produced. The density of the vehicle nodes is measured in per hour basis. Specifically we experimented with values of 1000 to 2500 with a step of 500, to imprint light and heavy traffic simulations, i.e., a sparse or dense vehicular network.

7.5.4.3 Velocity

For the speed of vehicles and with regard to an uban environment's restrictions, we draw a uniform distribution between 8-14m/s for each car that enters the simulation. Hence each respective vehicle has its own desired speed, which coupled with the different density values, generates a highly dynamic environment.

7.5.4.4 Neighborhood

The neighbor list for each vehicle is maintained by the periodic exchange of beacon messages. A typical beacon includes information about a vehicle's id, its position and speed. In our experimentation beacons are broadcasted every one second. To account for cases where messages are temporarily lost, e.g. due to building-obstacles, and not due to a car getting out of range, a vehicle removes a neighbor if it missed two consecutive beacon messages.

7.5.4.5 Virus Strength

Lastly, it is reasonable to assume that a virus may not be able to ”penetrate the defenses” of all vehicles it encounters [101]. This may be due to manufacturing aspects, antivirus flaws, etc. Thus, the virus is characterized by a final parameter, namely the *Virus Strength (VS)* indicating the number of vehicles in the simulation that are vulnerable to it. Hence, vehicles that cannot get infected, are set in the R state of the SIR spreading model, i.e., immune vehicles.

7.6 Results

Summarized in Table 7.1 are the parameters used in our simulation. Unless stated otherwise default values are used. Evidently when SHs have a broader scanning range, more vehicles are identified through the specialized hardware. In order to highlight the fact that the proposed method is efficient due to the dissemination of the lists (BL, PIV) among vehicles we keep the scanning range of the SHs to $30m$ for the entire simulation. Moreover, unlike static networks where the number of deletions is limited [27] [120], in a VANET nodes can be deleted in a broadcast fashion. Hence we choose to cut either all or half the neighborhood of an infected source as explained in subsection 7.4.2. Overall, the illustrated results are a fraction of the experimentation we conducted. In the current article we illustrate the most characteristic ones, nonetheless the qualitative conclusions are the same.

Table 7.1: Simulation Parameters

Parameters	**Range**	**Default**
Infection Delay (τ)	1 - 6	4
Vehicle Speed (m/s)	8 - 14	Uniform
Vehicle Density (per Hour)	1000 - 2500	1500
SH Scan Range (m)	30	30
Cut (%)	50 - 100	100
Vehicle Transmission Range (m)	300	300
Virus Strength (%)	25 - 100	100

7.6.1 Impact of Vehicle Density & Different Initial Spreader

This section evaluates the performance of the proposed technique as we increase in density, i.e., the number of vehicles. The results are illustrated in Figures 7.3 (by infection point) and 7.4 (by averages). When the diffusion process is in progress, higher density is interpreted in increased number of paths for propagating. This characteristic will pose significant challenges for any defense mechanism assigned to block the outspread of the infection. However, in our framework, these conditions will enhance the spreading of the virus negating elements as well, i.e., the BL and PIV lists.

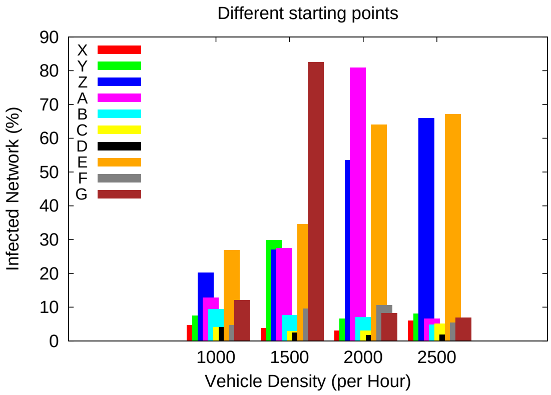


Figure 7.3: Percentage of the infected network from the different initial spreading points.

Figure 7.3 indicates that the road/traffic conditions that the infected vehicle experiences when the malicious propagation initiates plays a crucial role in the spreading dynamics; different number of one hop neighbors ranging from only a few to dozens; neighbors who co-travel for a long period or only for a few seconds; different speeds and directions between them etc. It is worth noting that the road topology (Figure 7.2) used in our simulation has a wealth of obstacles (buildings) which interfere with the communication of vehicles. Moreover there are several locations which favor the spreading process more than others. For example, Area 1 mostly allows spreading in a vertical or horizontal fashion. In Area 2 horizontal transmissions are often blocked. On the other hand in Area 3 or around the area of $SH1$, transmissions occur in all possible directions (horizontal, vertical, diagonal, etc.) due to the existence of large open areas, i.e., sparser buildings locations, providing a more favorable environ for the virus to propagate faster with respect to the other areas. Hence, it can be concluded that these network parameters, play significant role in the spreading of the virus and the diffusion dynamics of our defense mechanism.

Figure 7.4 shows that for sparse scenarios the infection is non-epidemic when we include 100% of an infected vehicle's vicinity in our PIV, i.e., the infected fraction is near 10%. Reminisce that the infection delay is four transmissions ($\tau = 4$). As more vehicles are introduced in the simulation, e.g., 1500-2000V/h, a larger fraction of vehicle nodes become infected, about 24%. However, as

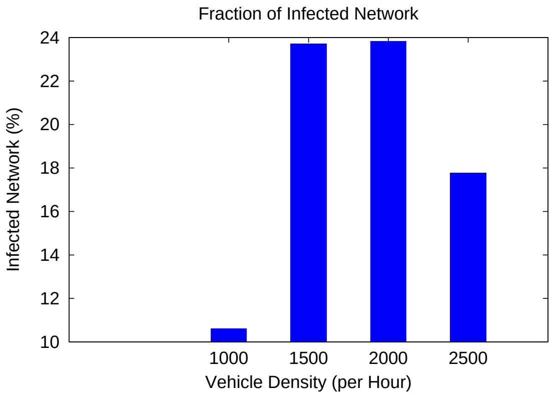


Figure 7.4: Average infected network size.

the evaluated scenarios become even more dense (2500V/h), more vehicles are included in the PIV list and thus the available paths for the worm-virus to spread, decrease. On the other hand vehicle paths for exchanging BL and PIV lists are only increasing and thus the efficiency of the proposed technique is enhanced.

7.6.2 Impact of Infection Delay (τ)

Next we investigate on the impact of the infection delay (τ). Evidently the increase in the number of necessary transmissions needed for the virus to propagate has positive impact on the proposed defense method. In other words, the longer it takes for the virus to travel from vehicle-to-vehicle, the more time we gain to circulate both, the PIV and BL lists within the vehicular network. Moreover the existence of obstacles will further delay the propagation of the virus, whereas the proposed technique will be less influenced since a single transmission is needed to inform susceptible vehicles.

As illustrated in Figure 7.5, when $\tau = 1$, i.e., when the infection is instantaneous between vehicles, the lost fraction of the vehicular network is near 80% as the proposed mechanism cannot “outrun” the malicious propagation. In such extreme scenarios any similar defense mechanism would prove inadequate to block the outspread of the virus. For $\tau = 2$ the diffusion of the virus is significantly mitigated through the proposed technique, whereas for $\tau = 6$ the infection is limited to only 10% of the vehicular network.

7.6.3 Impact of Virus Strength

In Figure 7.6 the x-axis represents the fraction of the network nodes susceptible to infection. The results illustrate that when the number of vehicles that are vulnerable to infection decrease, the virus propagation becomes more difficult. This is due to the fact that from the perspective of the virus, the network becomes more sparse and potentially disconnected. On the other hand, this

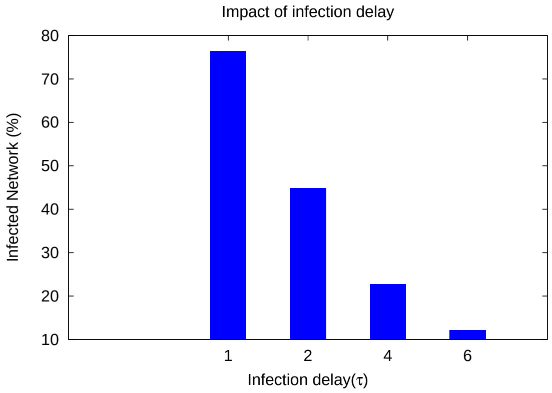


Figure 7.5: Impact of the tranmissibilty of the virus.

feature only affects positively the proposed method, since these ”firewall” nodes will hinder only the spread of the virus while the circulation of PIV and BL is left undisturbed.

As the spreading paths−for the virus−are gradually diminishing, the virus ”speed” is mostly based on the respective vehicle's velocity and the topological characteristics of the road network for overcoming potential disconnected vehicle paths. Under these circumstances, the ability of the virus to become epidemic is questionable. On the other hand, even when 100% of the network is vulnerable to infection, about 23% of the VANET is infected, which highlights the efficiency of the proposed mechanism.

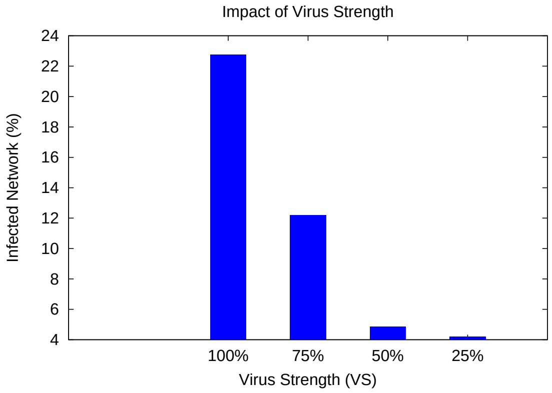


Figure 7.6: Vulnerability of vehicles to infection.

7.6.4 Impact of Different Cut Methods

In this section we evaluate the performance of the proposed method by employing a more elastic methodology for the neighboring vehicles of a newly identified infected source. Particularly, as described in section 7.4.2, we place nodes in PIV based on their connectivity (degree), i.e., the most connected nodes first, or their contact duration, i.e., oldest co-travelers first. Figure 7.7

illustrates the results when including 50% of each vehicle's neighbors in PIV. The x-axis depicts the different approaches.

Evidently, temporarily blocking nodes based on their connectivity yields better results. This is due to the very nature of the VANET; the existence of upcoming congested intersections, road segments of different priorities, traffic lights etc., resulting in a dynamic traffic environment where vehicles slow down or line up for arbitrary lengths of time. Thus, in such cases choosing nodes with respect to contact duration will be less efficient. On the other hand, by selecting (influential) vehicles in decreasing order of connectivity, i.e., locally more connected/central nodes, the proposed mechanism is found more efficient in blocking the outspread of the virus.

Among the various vehicles included in PIV, vehicles that are not truly infected are also present. Particularly, for the degree method (and default system parameters) we recorded that among 153 vehicles (on average) included in PIV, 30 vehicles were not truly infected. Although this is not a negligible portion of vehicle nodes, our results indicate that a more sophisticated cut method can reduce those "false positives" even more. Overall, moving vehicles in PIV means cutting communication paths for vehicles that may not be infected, which can result in additional delay on applications running on VANETs. Nonetheless this is only a temporal (but necessary) effect of the proposed technique, for efficiently blocking the outspread of the virus.

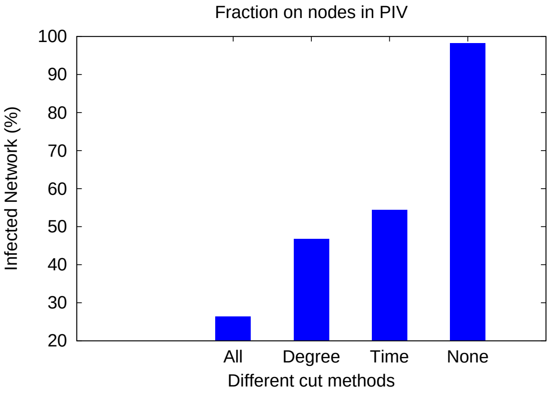


Figure 7.7: Cutting different neighbors from infected nodes.

7.7 Conclusion

This article proposed a distributed solution for hindering the outspread of a virus in vehicular networks by initiating a negating spreading process to counter the outspread of the malicious propagation. Inspired from complex network theory mechanisms, we introduce two competing spreading process in the vehicular environment where we try to shield vehicle nodes from a worm-virus, propagated through vehicle communications. Our simulation showed that the proposed mechanism significantly hindered the outspread of the virus even when the entire network was susceptible to infection. An interesting future direction resides in devising more sophisticated approaches for selecting quarantined neighbors, and furthermore “flexible” worms capable of adapting in countermeasures induced by defense mechanisms.

CHAPTER 8

PROTECTING A VEHICULAR NETWORK FROM INFECTED NODES

A Robust Eco-Routing Protocol Against Malicious Data in Vehicular Networks

8.1 Introduction

In this chapters we further discuss on the VANET ecosystem and routing protocols in urban environments. Of particular importance are environmental-friendly mechanisms, including the reduction of CO_2 emissions and mileage [3] [74], since vehicles not powered by fossil fuels will not be replaced soon, e.g., by fully electrical vehicles. The evolution of vehicles to mobile connected entities with On-Board-Units (OBUs) and Internet access [127] exposes otherwise legitimate vehicles to potential threats, i.e., infected with malware. Reports [4], [5] indicate that the infection of vehicles is now, indeed, a realistic scenario and the involvement of such in VANET protocols can result in catastrophic events. Examples range from injecting false data to disrupt the vehicular environment, e.g., with false data related to traffic congestion, traffic accidents and road conditions [32], to inhibiting communication, e.g, by jamming [44], or to more extreme phenomena such as endangering human lives by taking control of a vehicle [151].

In [98] we proposed a routing protocol, the eco routing of vehicles (ErouVe) mechanism, which utilizes vehicle-to-infrastructure (V2I), infrastructure-to-infrastructure (I2I) and infrastructure-to-vehicle (I2V) communications to provide routing instructions to vehicles, for a greener trip

Related publication **[C3]**: Pavlos Basaras, Leandros Maglaras, Dimitrios Katsaros, Helge Janicke. *A Robust Eco-Routing Protocol Against Malicious Data in Vehicular Networks*, **Proceedings of the 8th IFIP Wireless and Mobile Networking Conference (WMNC)**, pp. 184-191, Munich, Germany, October 5-7, 2015.

towards their destination, i.e., optimizing travel duration and CO_2 emissions. However, the original *ErouVe* algorithm, offers no protection against bogus information originating from infected vehicles and identifying potential vulnerabilities in a connected car's communication system, is a key factor for shielding it against rational attacks. As online attacks have become potentially more hazardous and aggressive in recent years, the development of real time defense mechanisms has been stepped up.

To this end, in the current work we focus on providing an effective defense system against potential spurious data "running" through the system's communication phases, which are aimed at disrupting *ErouVe*'s routing decisions. Our simulation results show that the proposed defense mechanism successfully identified outliers and restored ErouVe to near original instructions, i.e., as if no bogus data was present. An important information element in VANET communications is the position of adjacent nodes since most applications rely on them. Functions, such as the geographic routing on the network layer or the V2X applications, require genuine, accurate and reliable location data regarding neighbors. As a result, we propose to verify the consistency and plausibility of location-related data of adjacent nodes that are broadcasted frequently as Cooperative Awareness Messages (CAMs) or geo-networking beacons.

8.2 Related Work

Inter Vehicle Communications (IVC) support applications that are related to safety [168], traffic management [67] and infotainment, with most of these applications requiring frequent data exchange among vehicles. In addition to reassuring that packets are delivered on time, which is crucial for safety applications, mechanisms that ensure accuracy and consistency of the data are required. In order to provide a secure environment for vehicular communications we need to consider information security requirements, such as confidentiality, integrity and authentication. There are numerous kinds of attack that may threaten confidentiality, availability and authenticity of data [37].

Many routing protocols try to establish paths among entities that could provide fast and reliable communication. During the creation of these routes vehicles exchange information about their position, velocity, direction etc., and a mechanism is used to select those nodes that are optimal for each protocol. In a black hole attack, a malicious node exploits this mechanism by advertising itself, e.g., as a shortest path vehicle, to attract significant data traffic [108]. The attacker can choose to drop the packets or manipulate the data, for example by sending them to the wrong recipient. As a result, the source and the destination nodes become unable to communicate with each other. Denial of Service (DOS) and Distributed DOS attacks can affect the availability of the data, since the attacker can jam the medium, thereby disrupting the communication among the nodes. The authors in [44] showed that RF jamming poses a serious threat to safety in VANETs, for according to their experimental study, jammers can severely disrupt communication up to $465m$ despite very short communication distances between legitimate devices. During a Sybil attack [130], a malicious vehicle may pretend to be multiple vehicles and then use these multiple IDs to distribute false information. The deleterious effects of such attacks can cascade through the network and cause problems in the proper dissemination of information. Timing and node impersonation are two other example of attacks affecting the correct delivery of the information that can be easily launched in a vehicular environment.

A first step towards devising an appropriate defense system is the ability to detect infected vehicles. As noted in [33], misbehavior detection in VANETs can be divided into *Node-centric* or *Data-centric* mechanisms, with the first inspecting the behavior of a vehicle node, but not the data it sends. For example, if the rate at which a node sends packets exceeds a normal (predefined-historical) one, it is characterized as a misbehaving vehicle [37]. Other mechanisms in the same category include some form of reputation management, which inspects the past and present behavior of a node to derive the probability of future misbehaviour, as implemented in [115].

Filtering out false data is another technique widely used in wireless sensor networks and VANETs [158]. Our proposed scheme is based on a form of reputation and filtering, since vehicles constantly exchange their current information, which they use in order to create and maintain a list of their neighbors. In our defense mechanism, all the data collected from the vehicles are gathered and validated by the RSUs [1]. This way, information that is sent from infected vehicles is discarded and hence, their credibility is considered to be zero.

The second discrimination concentrates on the disseminated data in order to detect misbehaving vehicles, a scheme which is also used in our proposed defense system. Specifically, the disseminated data are evaluated for *plausibility* and/or *consistency*. For example in our evaluation scenario, plausibility will ensue if a vehicle reports a travel time of a few seconds while traveling a relatively long path. Consistency will be applied if a vehicle sends high (or low) statistics for a road segment, e.g. CO_2 emissions depending on the attack's goal, which although plausible, significantly deviate from similar reports of other nearby vehicles.

8.3 Preliminary Work, *ErouVe*

The original *ErouVe* algorithm was presented in [98]. The protocol identified traffic congestion phenomena in specific road segments, by taking into consideration the travel duration and CO_2 emitted by vehicles. In the sequence we describe the algorithm's specifications and functionality along with the new mechanism for routing instructions.

[1] http://www.bmvi.de/SharedDocs/EN/Anlagen/VerkehrUndMobilitaet/Strasse/cooperative-its-corridor.pdf?__blob=publicationFile

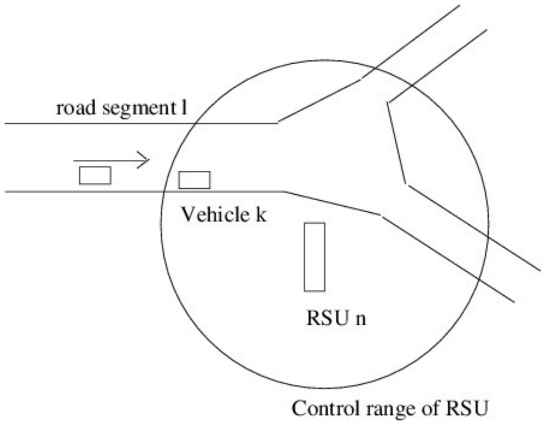


Figure 8.1: CO_2 emissions reduction system based on DSRC communications

8.3.1 System Description

We employ a network system $G = (V,L)$, where V depicts a set of nodes (intersections - RSU placements) and L are the road segments connecting those intersections. The set of road segments adjacent to an RSU is denoted as $S(n)$, $\forall n \in V$. RSUs with common adjacent road segments are considered as neighbors that we denoted as $N(n)$, $\forall n \in V$. Note that two neighboring RSUs may be connected through more than one route. Vehicles send data regarding a traversed road segment $l \in L$, to the corresponding RSU (Figure 8.1), including travel duration and CO_2 emissions. Following, neighboring RSUs exchange the respective information acquired from several vehicles and calculate average values for all adjacent road segments ($\forall l \in S(n)$). These values will project the vehicular environment for vehicles willing to traverse specific road segments. In order to have updated information for each road segment, a time window is introduced, namely time interval (TIN), from which a specific eco-route for each vehicle will be identified. Note that ErouVe runs on level 2 of automation[2] to advise upcoming vehicles; "Combined function automation".

8.3.2 System Initialization

In the initialization phase we build the network topology, that is, all RSUs become aware of their neighbors and their in between distance with respect to the road segments that connect those RSUs. Note that no time or CO_2 cost is initially calculated for the road segments. Table 8.1 briefly describes the initial information stored by each RSU. As illustrated, column 2 holds the neighbors of each RSU, column 3 has the road segment(s) through which neighboring RSUs are connected and finally, column 4 illustrates the distance for each road segment. Briefly, a vehicle k from R1 can reach R2 through segments l_a and l_b in distances D_a and D_b, respectively.

[2] http://www.nhtsa.gov/staticfiles/rulemaking/pdf/Automated_Vehicles_Policy.pdf

Table 8.1: Example of Connections Table for 3 RSUs

RSU_Id	Neighbors	Road Segments	Distance
R_1	R_2, R_3	R_2: l_a, l_b R_3: l_c	R_2: $l_a(D_a)$, $l_b(D_b)$ R_3: $l_c(D_c)$
R_2	R_1, R_4	R_1: l_a R_4: l_d	R_1: $l_a(D_a)$ R_4: $l_d(D_d)$
R_3	R_1, R_5	R_1: l_b R_5: l_e	R_1: $l_b(D_b)$ R_5: $l_e(D_e)$

8.3.3 Communication Phases

This section briefly explains the different communication phases of the original algorithm.

8.3.3.1 Road Segment Measurements (I2V)

For any vehicle k, which just completed its course on road segment l the corresponding RSU impels vehicle k to:

- calculate the travel duration (TT_{lk}) and CO_2 emissions (C_{lk}) on road segment l
- send to the RSU the respective values of TT_{lk} and C_{lk}

8.3.3.2 Communication of RSUs (I2I)

The communication between neighboring RSUs follows by sending the respective values of travel time and CO_2 emissions through beacon messages, for all vehicles that traversed the specified road segments. Each RSU averages those values to project the traffic conditions for each road segment, and select an appropriate route for each vehicle separately.

8.3.3.3 Route Request-Reply (V2I)-(I2V)

Each vehicle k that enters the control range (intersection area) of an RSU sends a route request message (R_q) to the corresponding RSU, which in turn, after solving the optimization problem (cf. next subsection) based on data obtained through I2I, sends routing instructions to the corresponding vehicle via an R_a message (route answer).

8.3.4 New Decision System for Route Selection

In the original ErouVe mechanism, as presented in [98], weights are assigned for each road segment adjacent to an RSU. The weight values are a combination of travel duration, CO_2 emissions and the additional travel distance towards a vehicle's destination. Finally, the route with the minimum weight is communicated through the R_a messages. By following a slightly different approach we developed a multiple decision mechanism, depicted in Figure 8.2. The new mechanism, rather than adding the different values of the three features used, i.e., travel duration, CO_2 emissions and distance, it logically combines the outcomes of three decision rules, each representing one of them respectively.

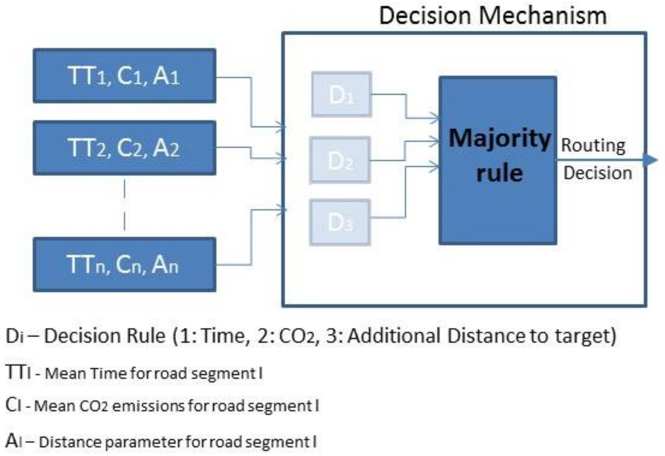

Decision Mechanism
TT1, C1, A1
TT2, C2, A2
TTn, Cn, An
D1
D2
D3
Majority rule
Routing Decision
Di – Decision Rule (1: Time, 2: CO2, 3: Additional Distance to target)
TTI - Mean Time for road segment I
CI - Mean CO2 emissions for road segment I
AI – Distance parameter for road segment I

Figure 8.2: New decision mechanism

In the new ErouVe mechanism, the RSU, after receiving a route request message from an approaching vehicle k, compares the respective road segments based on the current mean time, mean CO_2 and the added distance that each routing decision brings about. The outcomes of each decision are combined using weighted majority voting and different weights can be used in order to focus on one of the different optimization parts, e.g., time, distance or CO_2 emissions. In the default system settings, all optimization parts have the same significance. For example when comparing two potential routes, e.g., k and l, if D1 and D2 for k are greater than D1 and D2 respectively of l, l is selected as the next road segment.

8.4 ErouVe Vulnerabilities

$ErouVe$ utilizes V2I, I2I and I2V communications, in order to decide on which is the most eco-friendly route for any vehicle to follow. However, the technique's performance so far, assumes that vehicles will send only real data to the corresponding RSU. If we devise a scenario where *tweaked* information exists among the received data, the algorithm's formula can mislead vehicles to not only false eco-friendly routes, but also, create traffic congestion and significantly deteriorate the system's performance, i.e., increase travel time and CO_2 emissions.

In this study, we classify tweaked information into two basic categories depending on how an infected vehicle may attempt to manipulate data:

- Send tweaked data to *favor* a route (FAV)
- Send tweaked data to *fend* from a route (FEN)

FAV can be regarded as an attack that creates a false image for a specific road segment by sending relatively small statistics, i.e., short travel duration or minimum CO_2 emissions. For these scenarios, vehicles would be instructed to follow the *attacked* route, however, if the road capacity cannot satisfy the increasing number of vehicles, traffic congestion phenomena would emerge. FEN also alters the real conditions regarding the road segment under consideration but follows a reverse policy from FAV, i.e., by sending incremented statistics to the RSU, respectively. Hence, FEN, will direct vehicles towards different paths that could result in ambiguous traffic conditions.

Nonetheless, modified data regarding the accumulated CO_2 emissions or travel duration per road segment, is not the only vulnerability of the original *ErouVe* algorithm. Recall that once a vehicle exits the road segment under consideration, it sends a report to the corresponding RSU about the “condition” of the road segment it has traversed. However, so far RSUs have had no knowledge of which route the corresponding vehicle actually followed, apart to what was stated by the sending vehicle itself, and thus, cannot distinguish to which route the received data belongs. Consequently, an infected vehicle can denote that these values correspond to a different route (regardless of whether these values are altered or not) and hence, meddle with the system’s next decisions. With the above considerations, the original algorithm stands unprotected (vulnerable) to such false information and thus, our primary objective lies in devising a defense system to counter data originating from such malicious vehicles.

8.5 Attack Plans

8.5.1 Attack Objectives

To built on our defense system, we discuss several attack plans and their impact on *ErouVe*. The original *ErouVe* algorithm was implemented in order to balance the traffic flow between all possible available routes with a common destination. The proposed technique was compared to a scenario where the shortest route, followed by all vehicles, was unable to satisfy the traffic flow, thereby creating congestion in the path. By experimenting in high density traffic conditions, we found that *ErouVe*’s routing instructions successfully managed the traffic flow between the corresponding available paths and as a consequence, significantly enhanced the system’s performance, i.e., up to 30% improvement in travel duration. As a result, our attack plan focuses on sending “appropriated” (tweaked) data to recreate a scenario where all vehicles follow the shortest path and create congestion, although under the *ErouVe* paradigm. Intuitively, a combination of attacks, i.e., vehicles sending *favorable* statistics regarding the shortest road segment, i.e., FAV, and complementary *unfavorable* ones for the other route(s), i.e., FEN, will affect the systems routing decisions. By reversing the attack plan on the road segments, i.e., FAV for the longer routes and FEN for the shortest path, we obtain a different impact on the protocol’s routing decisions. In this scenario, vehicles will unnecessarily be rerouted to longer routes, resulting in increased travel duration and CO_2 emissions for each individual vehicle and concurrently, the system.

The aforementioned attack plans have contradictory objectives. In this study, we focus on the recreation of congestion for the shortest route by exploiting the vulnerabilities of the original protocol, i.e., Fake Route (FR) and Fake Data (FD).

8.5.2 How To Attack

First, recall that *ErouVe* uses data collected from vehicle measurements, accumulated within the most recent time window of s seconds (TIN), that is, bogus information has a maximum lifetime of TIN in *ErouVe*. Moreover, our experimentation showed that data from a single infected vehicle can have zero effect in the original *ErouVe* protocol, i.e., does not sufficiently change the weight values assigned to road segments and thus their overall ranking, although this is highly dependent on the extent to which the data are altered from their original values. However, if an attacker tries to use significantly deviated values to affect the formula/protocol, the received data from other (healthy) vehicles in a relatively short time, would render the identification of such bogus vehicles an easy task.

Since a single bogus vehicle may not make a difference to the protocol's routing decisions, grouped attacks are necessary, i.e., a number of infected cars that report their experience to an RSU for a target road segment in a relatively short time. However, bogus information has a lifetime of TIN in *ErouVe*, thus these reports must be defined with respect to TIN. As a final observation, on the occasion where a successful attack occurs, the system can still recover quickly if the weighted order of road segments is not changed much and a sufficient number of healthy vehicle reports follow. Consequently, catastrophic results, i.e., creating traffic congestion or unnecessarily rerouting a large number of vehicles to longer routes, can still be avoided, even with no sophisticated protection against false information.

To summarize, vehicles must not only meddle with the data to a degree that will not be undone with a few upcoming healthy vehicles, but also, to such an extent that it will not make the RSU suspicious, i.e., it cannot send extremely deviated values from the actual measurements. Finally, timed attacks are essential with respect to TIN as a single vehicle might not make a difference in the overall ranking of the road segments.

8.6 Proposed Defense System: Enhanced ErouVe

The goal of our defense system is to filter out false data, so as to return the functionality of *ErouVe* to near identical routing decisions, i.e., to an attack free scenario. Hence, data received by an RSU will be "judged" for both *plausibility* and *consistency* [33].

8.6.1 Fake Route Countermeasures

In order to counter the fake route problem we utilize the yet unused communication phase, i.e., *Vehicle-to-Vehicle* (V2V) communication in our model. To this end, vehicles traveling for instance on a specific road segment l, broadcast beacon messages regarding the vehicle's ID and that of their current road segment, e.g. l. Upon exiting the road segment under consideration, a vehicle k now sends information regarding not only TT_{lk} and C_{lk}, but also, the vehicle IDs that co-traveled with vehicle k on road segment l.

By instructing vehicles to gather information about their vicinity in their current road segment, bogus vehicles cannot state a different route than the actual one they followed. This is due to the fact that the current mechanism allows an RSU to have an accurate image for which vehicle followed which route based on the majority of votes. To bypass the system's new defense, a large number of infected vehicles need to be grouped appropriately, i.e., of magnitude greater than the currently healthy vehicles in the corresponding road segment. Nonetheless, in such a scenario, where the majority of vehicles are infected, all defense mechanisms are bound to fail. In our experimentation, we assume that beacons exchanged between vehicles cannot be "heard" in different road segments. This can be justified if we consider that the distance between the road segments could be greater than the standard DSRC communication range or because of the existence of obstacles, e.g., buildings in an urban scenario that interfere with the communication.

8.6.2 Fake Data Countermeasures

After properly matching data to the corresponding routes, we have to deal with vehicles that fake their accumulated statistics of travel duration and CO_2 emissions. First, we assume that statistics from healthy vehicles in short time, e.g., of a few seconds, cannot deviate significantly. It is a reasonable assumption if we consider that nearby vehicles will experience similar traffic conditions. Now, we need to clarify the validity of each newly received vehicle report. To this end, we define a new time window of about a third of TIN, to hold the reports for a set of vehicles in a very recent image of the road segment under consideration, namely *Validation Set Window (VSW)*. The Euclidean Distance between the report under "judgment" and those in VSW will decide the validity of the new data:

$$D(x) = \sqrt{\sum_{i=1}^{N}(x - y_i)^2} \tag{8.1}$$

where x stands for CO_2 emissions (or travel duration) of the new vehicle and y_i for the corresponding N values in VSW. $D(x)$ is compared to a threshold (TH_d) that determines its validity. However, a distant report is not necessarily a fake one, i.e., it may correspond to a true change in the traffic conditions of a road segment from dense to light traffic (congested to uncongested) and vice versa. Consequently, once a distant vehicle is identified, we do not take prompt action to drop its data, but rather save them in a separate set, namely, *Potentially Bogus Set (PBS)* in order to account for the abovementioned case. If $D(x) < Th_d$ then x ∈ VSW, otherwise x ∈ PBS. Parameter TH_d determines the sensitivity of the defense mechanism when categorizing new data as normal or fake, cf. subsection 8.7.4. We expect that if the report corresponds to a realistic traffic change, a number of similar ones are to follow. If the upcoming values are consistent with those in VSW, then the values in PBS are dropped and labeled as truly bogus data. Alternatively, if the size of PBS grows beyond that of VSW, we acknowledge a traffic shift and thus, integrate values of PBS to VSW. Figure 8.3 illustrates the proposed mechanism. Data are consistent (VSW) when below the threshold and otherwise inconsistent (PBS).

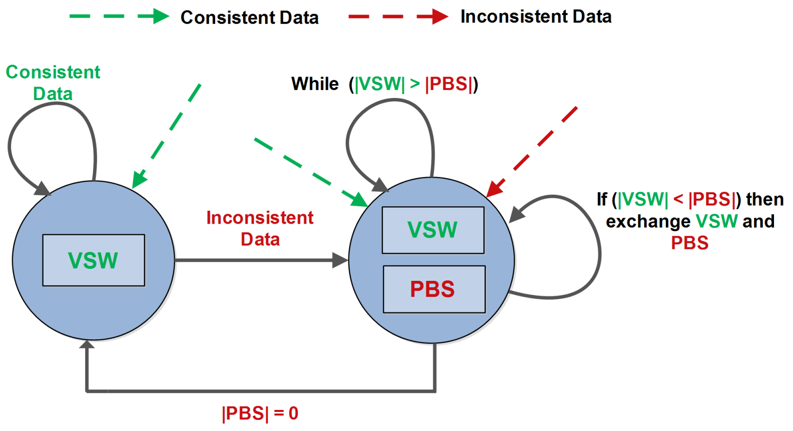


Figure 8.3: Fake Data Countermeasures

Finally, we should note that as explained in Section 8.3, a vehicle sends an R_q message in order to receive instructions. This places the following constraint: vehicles cannot easily lie about their travel duration. This is due to the fact that the RSU is aware of the time interval between the reception of an R_q message, and the time it receives the statistics from the corresponding vehicle. Nonetheless, more sophisticated plans can be deployed to fake travel duration, but are beyond of the purposes of the current study. Henceforth and without loss of generality we assume that only CO_2 emissions are altered.

8.7 Simulation Settings

8.7.1 Simulator

For the evaluation of our model, we use the simulator VEINS [129], which is composed of two well known simulators: OMNET++ an event-based network simulator and SUMO, a road traffic simulator. To calculate CO_2 emissions for each individual vehicle we apply the EMIT model integrated in VEINS. It is a statistical model for instantaneous emissions and fuel consumption based on the speed and acceleration of light-duty vehicles.

8.7.2 Evaluation Scenario

Similarly to our previous work [98], we built a map about $2km$ long (Figure 8.4) with a single direction and two available paths. The upper and longer path is about $275m$ long, whereas the lower and shorter path is about $190m$. Both road segments have the same capacity in lanes. These paths merge at junction 2, where the upper part can occupy 2 lanes of the next 3 lane road segment, whereas the lower part can occupy only 1. This setting is used to demonstrate a typical urban scenario, where part of a road can be temporarily closed due to maintenance or due to an accident. Another potential scenario includes crossroads with different priorities, where vehicles in the road segment with less priority line up and give room to traffic flows on roads with higher priority. Such considerations coupled with medium traffic can make a road segment that seems attractive, i.e., shorter path towards destination, unable to satisfy the traffic demand and consequently, result in major traffic congestion.

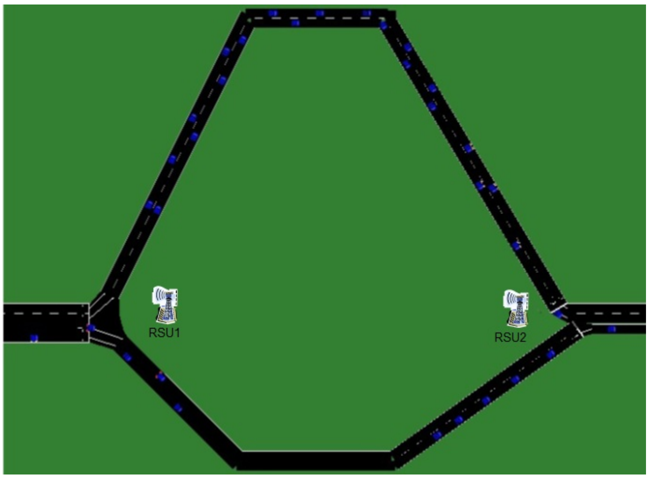


Figure 8.4: Simulation Map

8.7.3 Communication Settings

- **Communication Range**: this is the communication range that can be achieved between vehicles according to the setup of the system. In our experimentation it is set to $300m$.
- **Handshake Range**: at this range about ($100m$) an approaching vehicle is aware of the presence of an RSU and an upcoming intersection. This is facilitated through frequent beacon messages generated by an RSU. At this point, vehicles store the position of the corresponding RSU.
- **Control Range**: the final communication range of our system depicts the distance at which vehicles receive routing instructions (R_a message) from an RSU. In our simulation we set this range to a medium value, in order, if necessary, to give time to vehicles to perform rerouting, e.g., $50m$.

Table 8.2: Simulation Parameters

Parameters	Range	Default
Attack Type	FR, FD	FD
Group Size	1-5	3
Attack Interval (s)	6,10,14	10
FR Short Route	opt-2*opt	original
FR Long Route	opt-2*opt	original
FD Short Route	opt-2*opt	opt
FD Long Route	opt-2*opt	2*opt
Infected Vehicles (%)	10 - 30	20
TH_d (%)	10 - 50	10
Vehicle Speed (Km/h)	40 - 90	40
Number of Vehicles	50 - 150	150
TIN (s)	30 - 120	30

8.7.4 Parameters

In Sections 8.4 and 8.5, we elaborated on the vulnerabilities of the original ErouVe algorithm and devised attack scenarios to address those points. Table 8.2 summarizes the attack plans and their configuration: vehicle velocity, number of vehicles and TIN values as used in our experimentation. Group size depicts the number of consecutive vehicles that report false data, i.e., one to five vehicles, and attack interval is the interval between such groups, e.g., every six seconds. The attack intervals are chosen with respect to TIN, that is, at least two attacks groups must occur within one TIN. *opt* indicates how infected vehicles fake their original values in order to deceive the system. It is calculated for each road segment with respect to the road length and vehicle velocity, i.e, assuming vehicles travel in an uncongested road segment with the maximum allowed speed. For the FR attack, vehicles do not fake their reports, but rather, state that the accumulated statistics correspond only to the long route. For FD, bogus vehicles traversing the short route will say that they have experienced favorable road conditions, i.e. opt, whereas for the long route vehicles will state that there is significant congestion. Both attack protocols favor the short route in hopes of creating traffic congestion. Extensive experimentation was conducted in relation to the simulation parameters and in the next section, we present the most characteristic results. Unless stated otherwise, default values are used.

8.8 Performance Evaluation

8.8.1 ErouVe VS Shortest Path VS FR attacks

In Figure 8.5, the CO_2 emissions (ml) and travel duration (sec) of each vehicle are demonstrated. ErouVe in an unprotected mode performs similar to the original shortest path scenario, since due to the fake route attack it sends most of the vehicles to follow the lower road segment (shortest path). The increased traffic leads to road congestion that has an immediate effect on both travel duration and CO_2 emissions. That is, the mean increase in time and CO_2 compared to that in the attack free scenario is 31% and 20%, respectively. Such an increase can be further explained considering that ErouVe sends 25% of the vehicles to follow the longer route, whereas in the FR scenario only about 8% of the vehicles take the longer path. Such observations justify the need for countermeasures and as it will be illustrated, the proposed defense mechanism makes ErouVe robust to such attacks.

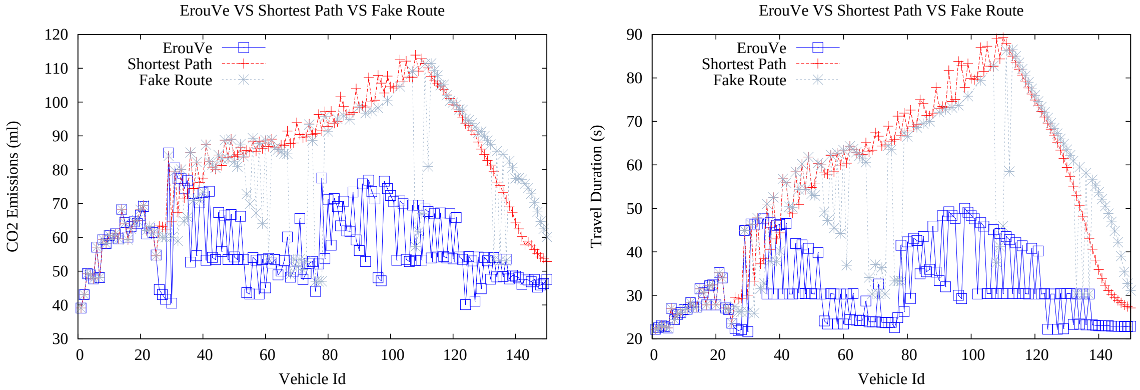


Figure 8.5: FR successfully deceives the original algorithm into sending vehicles to the short route and thus creating congestion. Travel duration and CO_2 emissions are significantly increased by 31% and 20% respectively.

8.8.2 Impact of Attack Group Size

Figure 8.6 illustrates how the number of consecutive vehicle attacks (attack group size) affects the system's average performance. The attack interval is set at 10 seconds. The Y-axis represents the deviation from an attack free scenario, i.e., depicts the performance drop. For one vehicle per 10 seconds we observe a minor deviation, for example, lower than 5% in CO_2 emissions. As the attack group increases and thus more fake data are running the system, the unprotected ErouVe mechanism is further deceived, e.g., more than 25% increase in travel duration for five vehicles per attack group. It is worth noting that one attacker per 10 seconds depicts 8.6% of 150 vehicles, while for a group of five vehicles, the bogus community rises up to 30%. Although this observation indicates a strong point for ErouVe because it takes a large number of vehicles to drop its performance about 25%, it also highlights the necessity for a defense mechanism capable of spotting spurious data to “cure” the system.

8.8.3 Impact of Attack Interval

In Figure 8.7, we investigate on the frequency of the attacks with the attack group size set to three vehicles. Note that zero in the x-axis represents the scenario with no fake data. As

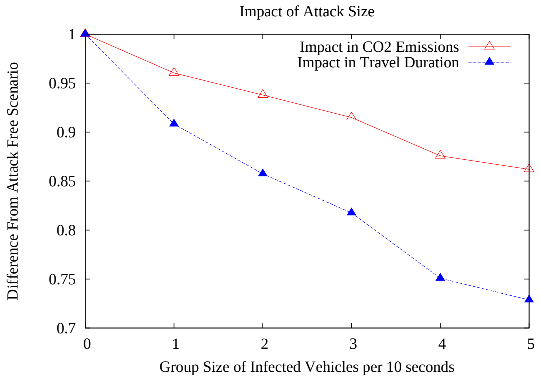


Figure 8.6: As the number of FD attacks running in system increases, ErouVe's performance drops. About 30% of vehicles out of the total simulation were bogus (attack group size set to 5) for a 25% decrement in travel duration.

illustrated, more frequent attacks have greater impact on the performance of ErouVe, e.g., about 24% increase in travel duration when attacks happen every six seconds, whereas there is 15% performance drop when the interval is 14 seconds. Note that for the interval of 14 seconds, only two attack groups "fit" in TIN, which explains the lower impact in the protocol's performance, i.e. false reports are not sufficient to change significantly the overall ranking of the road segments. As the simulation time flows, the impact of earlier fake data expires and consequently if no new such data arrive in short term, the system is very likely to recover to near normal routing decisions.

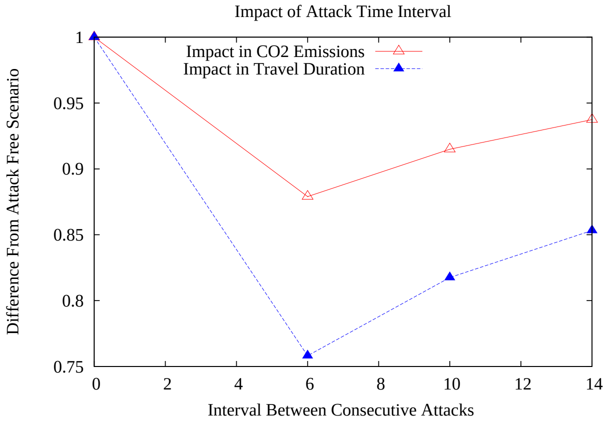


Figure 8.7: In order to significantly affect the routing decisions of ErouVe, fake data need to arrive in a timely manner, so as to continuously have false data in the system. Otherwise ErouVe may quickly recover to original routing instructions.

8.8.4 Impact of Defense System VS FD attacks

Finally, we present the performance of the proposed defense system against FD attacks. Recall that our goal is to have a performance similar to that of a scenario where no fake data are running through the system phases, and thus, illustrate the robustness of our defense mechanism. Figure 8.8 illustrates the obtained results. Evidently, the proposed method remarkably follows the performance of the original ErouVe algorithm. This is due to the fact that fake data are successfully omitted from the system, that is, ErouVe's routing instructions are only guided through the real traffic condition. The fraction of vehicles sent to the longer route is 27% for the defended ErouVe and about 18% when the defense mechanism in not active (vulnerable).

The deviation observed between the defended and original algorithm can be explained by the following reasons: first, since false data arrive in groups, i.e., three consecutive vehicles, when labeled fake and thus omitted from the system, ErouVe is left with no new received reports for an interval between the last received bogus data and the most recent true report. Second, a similar delay is induced in the protocol when data appears to be bogus, but it really is not, representing a traffic shift, between the time the report is labeled as BPS and later integrated in VSW. Such considerations induce a delay in the routing decisions and consequently, a deviation from the original ErouVe, but nevertheless are essential in order to filter out malicious vehicles.

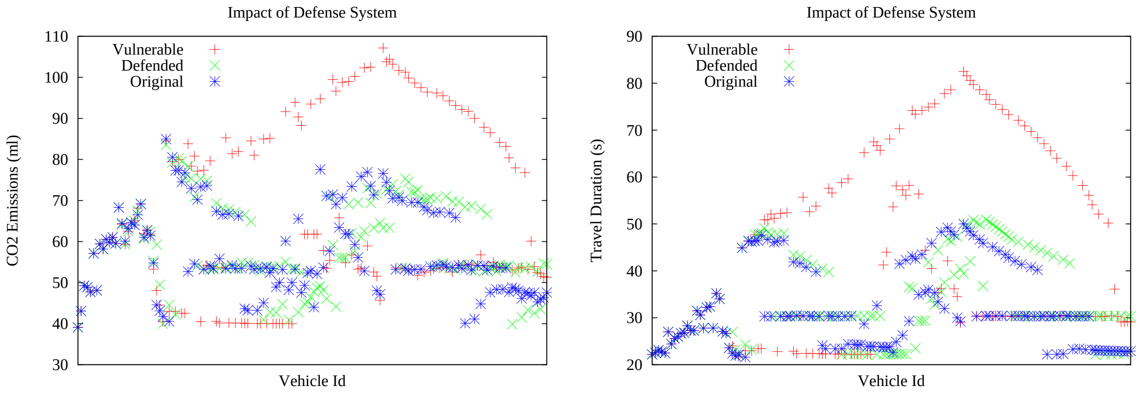


Figure 8.8: The proposed defense system returns the protocol to near identical routing decisions by successfully filtering out the outliers and thus the overall system's performance is preserved.

8.9 Conclusion

In this paper we investigate on how an eco-routing mechanism (ErouVe) that is based on DSRC communications, is affected from fake information disseminated from infected vehicles in an urban environment. We devised a set of attack plans with aim to guide vehicles to a "desired" route, and recreate traffic congestion. Subsequently we employed a defense methodology that relies on V2V, V2I, and I2I communication, that successfully filters out fake data running through the systems communication phases, that is, restore the performance of ErouVe to near normal operation. In the future, different attack scenarios are going to be investigated and more complex defense mechanisms developed.

Part IV

Low Cost Sampling Methodologies Based on Social Driven Aspects

Chapter 9

On neighboring nodes' relative power of influence

This Chapter focuses on understanding the connection between the influence power (and centrality) of relatively close neighboring nodes, by utilizing a social driven property. As we noted in earlier Chapters, the concept of identifying influential spreaders in complex networks has received increased attention in the past decade. A common characteristic for many of these network statistics deployed, is that the influential nodes they detect tend to be neighboring nodes. For instance, in highly assortative networks, high-degree nodes (i.e., hubs) are usually neighbors. Similarly, nodes belonging to the same k-shell are quite often neighboring nodes [36]. This characteristic has the consequence that the selection of such nodes as influentials might be redundant since the network parts that they can infect are highly overlapping. Also, the computation of many of these techniques or the selection of non-neighboring 'seeds' requires knowledge of the whole network topology; however more often than not we don't have the whole picture available, but only local information. Therefore, the need to compare the influence power of neighboring nodes arises; in other words, we need to answer a question of the kind whether the (direct or close) neighbors of a node are more influential than the node itself.

The answer to this question can straightforwardly be used for designing better influential nodes detection algorithms, or for estimating the spreading capability of nodes using their friends' capability; however, the question has an intellectual value by itself also, due to its relation to the well-known *friendship paradox* [193]. The phenomenon comprises an observation that, statistically, most people have fewer friends than their friends have. The last years, there have been some research efforts that investigated the friendship paradox relatively to some node 'quality' feature, e.g., with respect to prominence in science [52], popularity in Twitter [24], [89],

Submitted work [**S1**]: Pavlos Basaras, Giorgos Iosifidis, Dimitrios Katsaros, Leandros Tassiulas.*On neighboring nodes' relative power of influence*, **Submitted for journal publication**, October 2017.

happines [16], and so on. These studies of the so-called generalized friendship paradox [31], [52], [56] relate node features to inter-nodes links, and therefore relate nodal characteristics to network topology. However, the node characteristics that were investigated in [52] were all related eventually to the number (i.e., node degree) and type of coauthors. In scientific collaboration network and due to the way research is conducted, we (almost always) encounter the pattern that junior researchers, i.e., MSc/PhD/post doctoral students are cooperating (due to graduation, employment reasons) and thus coauthoring with more experienced researchers, namely junior and/or senior faculty, senior industrial personnel, and so on. These experienced researchers have (usually) more coauthors, more citations, more publications than their junior researchers; in other words, they are 'hubs' in the coauthorship network. Combining this with the fact that junior researcher population is larger than the seniors' population, we easily deduce why studying the 'paradox' in this way is not radically different than studying it in its plain form [193]. Similar arguments hold for the other aforementioned works related to the generalized friendship paradox. Therefore, that studies can not provide a clear picture about whether such generalized paradox holds in general.

In light of above discussion, our investigation has as its ultimate goal to settle the following question: *Are my (close) friends more influential than me?* By casting our investigation in the context of the generalized friendship paradox, we need to make clear the peculiarities of our study which make it interesting, and different than the study of other generalized friendship paradoxes. Firstly, the influence propagation is a probabilistic phenomenon and goes far beyond simple arithmetics (counting the number of my friends versus the number of my friends' friends, counting the number of my citations versus my co-authors' citations, counting the number of my re-tweeted tweets versus the number of my followers/followees' re-tweeted tweets, and so on). It depends on the spreading model and on its parameters, and while the tranditional friendship paradoxes refer to static centrality measures, our study involves a dynamic process. Secondly, the analysis of this 'paradox' might need no examination of all the nodes of a complex network, but only of those prominent nodes whose identification depends on the measure used, e.g., centrality, k-shell. Thirdly, many of these (generalized) paradoxes are explained by the fact that too many nodes are linked to a few hub nodes (or are co-authors of a few star-scientists in the case of scientific collaborations); in our study though, such explanation might not hold because it has already being proven in [139] that there is no strong positive correlation between node degree and influence capability, i.e., higher degree nodes are not necessarily better spreaders.

9.1 The influence power of my close neighbors

The vast majority of literature on influential node selection assumes that the whole complex network is available beforehand, and therefore it can be processed to infer the topological properties of each node, and then to compute network quantities such as centralities, cores and so on. Then, armed with a ranking of the nodes according to such a measure, the top-most such nodes can be selected as seeds to initiate a diffusion process. However, this methodology is only a part of a larger investigation effort, which has been overlooked so far. A thorough investigation of the topic should enlighten us on the following issues:

- Given a specific spreading model, characterize the spreading power of each node relative to the spreading power of its close neighbors. The investigation should go beyond the examination of 1-hop neighbors, and further examine its 2-hop, and 3-hop, and even more distant neighborhood if the finding call for such an investigation. The understanding of this issue is crucial for sampling purposes [136], [181]. It is also significant when the network is acquired in a streaming fashion, and we do not have the luxury of time to wait for its full topology or when the (main memory) storage capacity is inadequate.

- Given a specific ranking measure to sort the nodes (e.g., PageRank, k-shell, PCI, degree), characterize the relation of each node's value (for that measure) to that measure's value of its close neighbors. Again, this is significant for large scale networks where we cannot calculate this measure of all nodes (either it is computationally challenging, or it is time consuming, or it we do not have the necessary information at our disposal) and we need to use it in order to drive the selection of top-most influential nodes as required by methodologies found in the literature so far.

Even though there is no evidence in the literature, for instance, that each blogger believes that his/her posts are more influential [123] that the posts of his/her peers, we will take the liberty to introduce the term *influential spreaders paradox* to describe the phenomenon that, statistically, the spreading power of a node is inferior to that of its close neighbors. Similarly, we will introduce the term *centrality paradox* to describe the phenomenon that, statistically, the centrality value of a node is lower than that of its neighbors. The friendship paradox [193] is afterall a degree-centrality paradox. We will study these paradoxes both at individual node and at network level [193] (for their exact definitions see the section 'Materials and Methods'). We will say that the centrality (spreading) paradox will hold for an individual node if the node has lower centrality value (reps. spreading ability) than the average centrality (resp. spreading ability) of its (close) neighbors. On the other hand, we will say that the centrality (resp. spreading) paradox holds for a network if the average centrality (resp. spreading ability) of nodes is smaller than the average centrality (resp. spreading ability) of their (close) neighbors.

We admit that it is not possible to examine neither the applicability of the centrality paradox for each one of the hundreds of centrality measures that have been proposed so far nor the applicability of the spreaders paradox for each one of the tens of spreading models that have appeared in the literature. We will do it only for the two, most prevalent spreading models, namely SIR and SIS (see Appendix A.1), and for the most widely used centrality measures (see Appendix A.2), namely, degree (DEG), betweenness (BC), closeness (CC), PageRank (PR),

k-shell (CORE), a variation of it namely, onion spectrum (ONION), and an hybrid named Power Commuity Index (PCI) [77].

In summary, the contribution of this work can be summarized as follows:

- we investigate numerically, using typical network models and also actual instances of various networks, if the friendship paradox appears in various centrality measures, and then, we extend the idea to 2-hop and 3-hop neighborhoods and again investigate how the paradox effects evolves. Thus, we answer the following question: *"Are your close neighbors more central than you are?"*
- we focus on the paradox effect for the metric of influence, in various networks, and under various influence spreading mechanisms. Thus we answer the following question: *"Are your close neighbors more influential than you are?"*
- we develop a sampling method for the selection of influentials and for the blocking of contagions.

9.2 Results

Table 9.2 in section 'Data description' of 'Materials and Methods' describes the real networks used in our study; we used one communication network (Email-$Enron$), five co-authorship networks (CA-$Astroph$, CA-$CondMat$, CA-$HepPh$, CA-$HepTh$, CA-$GrQc$), three social networks ($Brightkite$, $Facebook$, $Hamsterster$), and one interaction network (PGP). For those networks composed of many connected components only the largest component is considered. We have also generated random networks using the R Project for Statistical Computing [11]; an Erdos-Renyi [15] network where the probability p for drawing an edge between two arbitrary nodes is set at 0.004, and a graph that follows the Watts-Strogatz small world [15] model with rewiring probability r at 0.4. The results illustrated for the artificial graphs are averaged over ten independent networks. In Table 9.2, $\langle \mathbf{k} \rangle$ depicts the average network degree, **D** corresponds to the network diameter with the 90-percentile effective diameter appearing in brackets, and finally **A** illustrates the degree assortativity. For more details, readers are referred to [8], [65]. We will focus on the extensively studied $Enron$ email network, and provide the results for the remaining networks in the 'Supplementary Material' of this article.

As usual, we model each network as a graph $G = (V, E)$, where V is the set of nodes, and E is the set of links among nodes; the cardinality of $|V| = n$ is the number of network nodes. We will denote the set of α-hop neighbors ($\alpha = 1,2,3$) of node i as $N_\alpha(i)$, and this set will include all network nodes that are within exactly α hops distance from node i, and with $|N_\alpha(i)|$ we will denote the cardinality of this set. We will define the spreading power of a node i, $SP(i)$, as the number of network nodes that they get infected when the infection starts from node i (see also section 'Spreading models'). We will first investigate the centrality paradox at both the network and the individual level followed by our results regarding the spreading paradox, and finally close our work with applications based on the article's findings.

9.2.1 The centrality paradox

Despite the thorough work done by the seminal studies reported in [193] and [52], their common feature is that they have both examined node 'features' that eventually reduce to merely counting links incident to a node. Departing from their perspective, we examine here several centralities some of which are known not to be strongly correlated to node degree, and secondly, we investigate whether the paradox holds for more distant neighborhoods of a node rather than just for its direct neighbors.

9.2.1.1 Centrality paradox at network level

We will start by examining whether the paradox holds at the network level which is a 'macroscopic' observation, and also by confirming a trivial result, i.e., if there are no 'hub' nodes which are responsible for creating the variance in the average of the measured quantity [193], then the paradox can not hold. Figure 9.1 illustrates whether the paradox holds at the network level for two classes of networks. The first class includes two networks that are knowed to exhibit power-law degree distribution, namely the real Email-Enron network (the first plot) and the artificially generated network following the Barabasi-Albert preferential-attachment generation model (second plot). The second class includes two networks with Poisson-like degree distributions, namely a pure Erdos-Renyi network (third plot), and a network which follows the Watts-Strogatz small-world model (fourth plot). The y-axis depicts the distance (in ratio) between the average centrality $\langle v \rangle$ of a node and the average centrality of neighbors $\langle v_{nn} \rangle$.

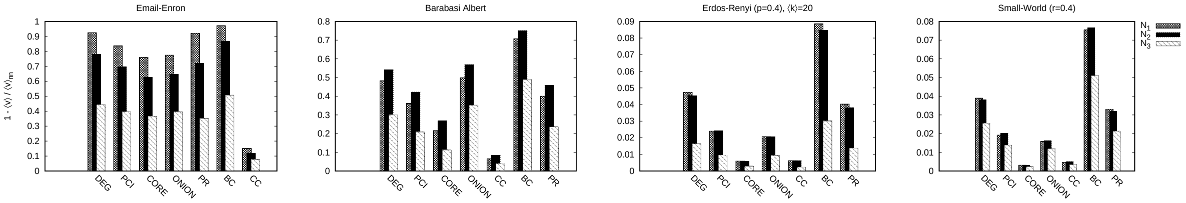


Figure 9.1: Centrality paradox at network level. The x-axis shows the evaluated centralities measures while the y-axis illustrates the distance in ratio $1 - \frac{\langle v \rangle}{\langle v_{nn} \rangle}$ for all neighborhoods (N_1, N_2 and N_3). The paradox holds for networks with power-law degree distribution due to the existence of hub nodes, but not for networks with Poisson-like degree distribution. The strength of the paradox weakens only for the N_3 neighborhood, whereas for the N_1 and $N2$ neighborhoods is very strong and in a way competitive way among them. The observation that the paradox apprears stronger in N_2 for the simulated Barabasi-Albert network is not unrealistic since it is observed in the *CA-CondMat* network.

The generic (expected) observation is that for power law networks the paradox is valid across neighbohoods. In particular, it is strong for N_1 and N_2, and it significantly weakens for N_3. This weakening trend for N_3 is characteristic in the majority of the evaluated networks (see Supplementary Figure C.1). In many of the cases, the probability of the paradox holding in N_1 is no more than 10% higher than the respective probability in N_2, and there is even the case of *CA-CondMat* network (Figure C.1) where the situation is reversed; this provides a significant first evidence that the paradox is not an oddity of the N_1 neighborhood only. When examining the validity of the paradox across centralities, we see that the paradox holds for all of them, with the exception of closeness centrality. The power-law distribution of the centrality values e.g., DEG [15], BC [172], PR [166] provides a rational explanation for this observation. The paradox holds with high probability even for CORE which has not an established power-law behavior. Closeness centrality is a departure from this rule and this is explained by the nature of this centrality measure; a really large number of nodes is located near the 'center' of any network when it is relatively densely connected, and this destroys the power-law behavior.

On the other hand, for networks with Poisson-like degree distribution the calculated probability value is always below 0.1, i.e., the paradox (as expected) does not hold and the main reason is the absense of hub nodes.

In summary, the centrality paradox at network level holds strong across centralities, and across N_1 and N_2 neighborhoods. We now proceed to study it at the individual (or node) level.

9.2.1.2 Centrality paradox at individual node level

Rcall that the centrality paradox holds for a node at the individual level if the node's centrality value is smaller than the average centrality value of its neighbors. We will use the symbol $h^{\kappa}_{\gamma}(s,v)$ to define the centrality paradox holding probability that a node with N_γ ($\gamma = 1,2,3$) neighborhood's size equal to s and κ (κ = DEG, PCI, CORE,...) centrality's value equal to v satisfies Equation 9.1. For instance, $h^{PCI}_{3}(150,15)$ represents the centrality paradox holding probability for a node whose PCI value is equal to 15 and its N_3 neighborhood contains 150 nodes. If some of the factors in this probability symbol e.g., centrality measure are left undefined, then we will use the _ symbol in their position, i.e., $h^{-}_{N_1}(\_,v)$. Figure 9.7 plots this probability for the *Email-Enron* network, for most of the centrality measures and N_1,N_2,N_3 neighborhoods. The rest of the plots are included in the series of Figures C.2 to C.10. So, each plot depicts the centrality paradox holding probablity for pairs of node neighborhood's size and centrality value. The color is analogous to that probability ranging from black (not holding) to yellow (holding).

The leftmost plot in each line of plots inf Figure 9.7 and in Figures C.2–C.10 examines the paradox's truth for the node's direct neighbors, and are analogous to the plots of [52, Figure 1]. It has been established [52], [193] that the paradox holds strong for degree centrality (DEG), especially for the lower degree nodes. We confirm that findings, and generalize them in the following way: *for a fixed neighborhood size, the centrality paradox holding probability decreases with increasing centrality value, for any centrality measure, and for all close neighborhoods*.

However, we there are three interesting observations in our study that were not documented in previous works. The first one is as follows. Earlier works established that the paradox holding probability takes all values from 1 to 0 for many different sizes of the N_1 neighborhood, and becomes 0 only for the largest N_1 neighborhood. Here, we establish that for some centralities, namely PR, BC and CC this behavior is 'binary', i.e., the centrality paradox either holds or not, no matter what the size of the neighborhood is. For the rest of the centralities, namely DEG, PCI, CORE, ONION this binary behavior is observed only when the size of the N_1 neighborhood becomes quite large. For instance, PCI and CORE illustrate a paradox holding probability of about 0.5 at the relatively low values of N_1.

The second and more striking observation is that this binary behavior is more evident in N_2 and even more evidently in N_3. This phenomenon can be (partially) explained by the following: $|N_3| > |N_2| > |N_1|$ for the majority of the network nodes. In other words nodes will be "compared" with more neighbors for their centrality index (with respect to N_1), which increases the probability of finding highly central nodes (e.g., hubs) and thus expose neighbor superiority. In other words, it is like having a uniform sampling process and a focal node with high centrality value, and asking what is the likelihood that this sampling process will select a N_1 (N_2,N_3) neighbor of the focal node with higher centrality.

Finally, a third departure from study [52] which showed that the node with minimum centrality value min is most likely to have neighbors with higher centrality values and thus leading to $h_{N_1}^{DEG}(\_, min) = 1$ is confirmed in our experiments (the bottom-most points in every plot are yellow colored), with the exception of CORE, where the centrality paradox holding probability may vary from 0.5 to 1.0 because of this centrality' definition and the network assortativity.

9.2.1.3 Summary on the centrality paradox

As a summary of the investigation of the centrality paradox we can establish the following strong results: a) at network level, the paradox holds strong in both N_1 and N_2 neighborhoods for (almost) all centrality measures for all power-law networks, whereas it does not hold for networks with Poisson-like degree distribution; b) at individual node level, for a fixed neighborhood size, the centrality paradox weakens with increasing centrality value, for any centrality measure, and for all close neighborhoods; c) at individual node level, the centrality paradox either holds strong (i.e., its probability is equal to 1) or it does not hold at all (i.e., its probability is equal to 0) for neighborhoods others than N_1, and this binary behavior is evident even for N_1 for PR and BC centralities; the roots of these observations at individual level are the power-law distribution of the centrality values and the network assortativity.

9.2.2 The spreading paradox

In the previous section we established centrality paradox along with several new facts related to it; this paradox might not seem that paradox after all considering that a 'deterministacally' computed quantity of a node is on the average lower than that of its close neighbors, given the power-law degree distribution and the assortativity of the network. However, influence propagation comprises a completely different situation because it involves a probabilistic diffusion process.

We will study the paradox at network level first, and then at the individual node level for the two prevalent diffusion models, namely SIR and SIS. In both SIR and SIS, the spreading rate λ of the difussion process is set near the epidemic threshold of each network (see Table 9.2) as it is broadly used for the identification of influential spreaders (e.g., see [139]). In SIS, the probability γ for returning from the infected (I) state to the susceptible (S) state is set to 1 which represents the worst case scenario for the SIS spreading model, and it is also not in favor of the spreading paradox's validity confirmation, i.e., the presented results comprise a 'lower bound' of the paradox's validity.

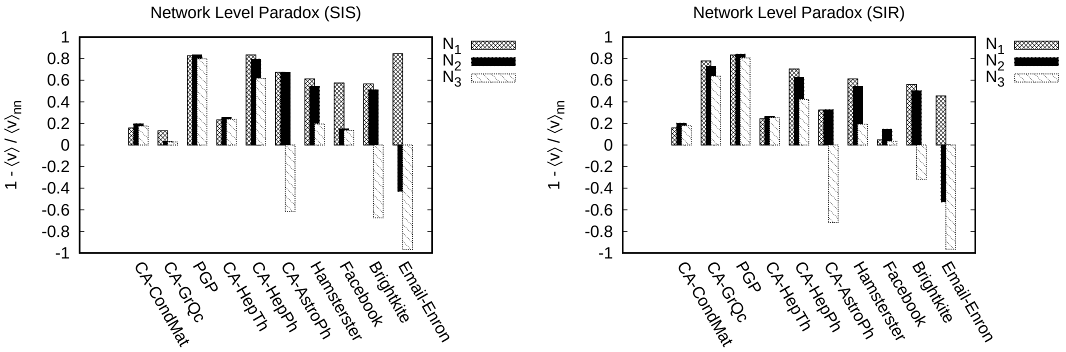


Figure 9.2: Evaluation of the influential spreading paradox at network level for the SIS and SIR spreading models in the *Email-Enron* network. The spreading paradox holding probablity is pretty high for the SIS model closely followed by SIR for the majority of the networks. The slightly lower paradox holding probablity for SIR is attributed to the existence of the R-state in that difussion model. Exceptions where the paradox does not hold are some very sparsely connected networks. The paradox holding probability is high in both N_1 and N_2 neighborhoods, which is a result observed for the centrality paradox as well.

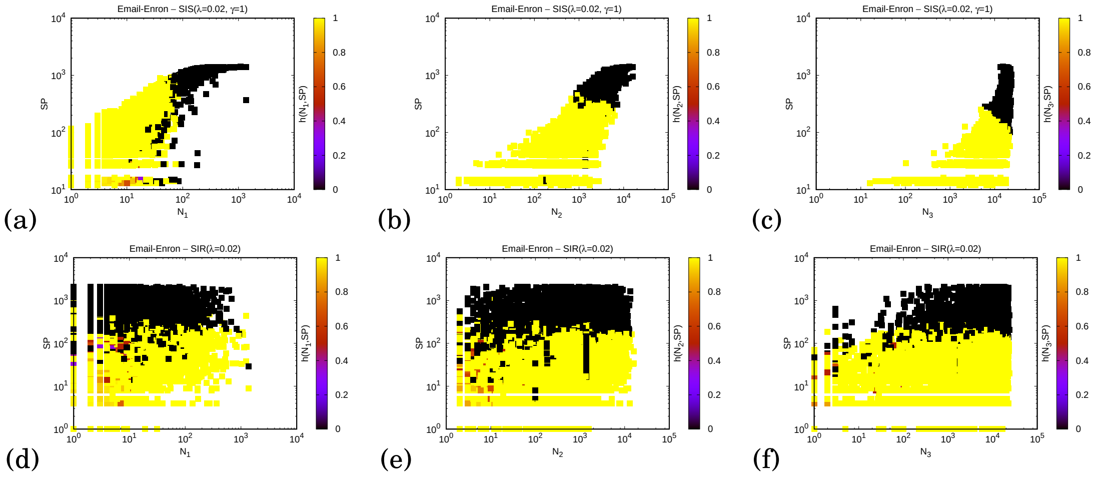
9.2.2.1 Spreading paradox at network level

Figure 9.2 illustrates the network level spreading paradox for SIR and SIS. The y-axis depicts the distance (in ratio) between the spreading power of a node and the average spreading power of its neighbors. At first glance it might seem that the spreading paradox holds but not strongly at network level. However this is not the case for the following reasons. In most network cases, the paradox holding probability exceeds 60% in N_1. The cases where this probability is below 20% for both SIR and SIS are for the networks *CA-CondMat* and *CA-HepTh* which are very sparse networks. Otherwise, the paradox holds strong and in fact it apprears quite strong in N_1 and N_2, just like the centrality paradox; in other words, *the friends and the friends' friends of a node are* *on the average better spreaders* than the node itself. The paradox weakens considerablly in most cases and even takes negative values when considering N_3 neighborhoods. When contrasting the spreading paradox holding probability in SIR and SIS, we observe that this probability is in general higher in the latter case, because SIR encompasses the R-state that reduces the size of the infected population. The real gap is even broader in favor of SIS, recalling that the SIS results represent a 'lower bound'.

9.2.2.2 Spreading paradox at individual node level

If taking averages (or even medians) of the spreading efficacy of nodes in probabilistic diffusion processes across the whole network (i.e., at network level) might result in smoothing out some particular behavior, the examination of this behavior in a local level (at individual node level) will remove any doubts. So, in Figure 9.3 we illustrate for the *Email-Enron* network the spreading paradox at individual node level. The colored palette illustrates the paradox holding probability for pairs of neighborhood's size (x-axis) and spreading power (y-axis) for all three neighbors, namely N_1, N_2 and N_3. The top row of plots is about the SIS model, and the bottom row about the SIR model.

The generic pattern that we observe is the following: *for a fixed neighborhood size, the paradox holding probability decreases abrupty from the 'holding state' to the 'non-holding state'*. This is true across N_1,N_2,N_3 and spreading models SIR and SIS. For the SIR, it is evident that this abrupt change happens when the spreading power exceeds a threshold value which is practically independent of the neighborhood size. This behavious is observed for the SIS model and the N_2 and N_3 neighborhood, but for the N_1 the threshold depends on the neighborhood size.



(a) (b) (c) (d) (e) (f)

Figure 9.3: Evaluation of the influential spreading paradox at individual node level for the SIS and SIR spreading models in the *Email-Enron* network.

9.2.2.3 Summary on the spreading paradox

As a summary of the investigation of the spreading paradox we can establish the following strong results: a) at network level, the paradox holds strong in both N_1 and N_2 neighborhoods, and it is stronger under the SIS spreading model; b) at individual node level, the paradox holds strong as long as the spreading power does not exceed a specific threshold, and this behaviour is valid for both SIR and SIS models.

9.3 Applications

We aim to evaluate the importance of our findings within the concept of spreading dynamics. We will follow the paradox intuition for accelerating a spreading process or hindering the outspread of misinformation in networked populations. First we will explain the mining mechanism for selecting the cascade initiators/blockers and then focus in the respective use-cases.

9.3.1 Mining Cascade Initiators/Blockers

In order to mine highly central nodes we follow a similar policy to [52]. Specifically, an initially random set of nodes is selected. Meaningful conclusion can be drawn only when the size of the selected set, namely RND, is relatively small, hence we experimented with 10 and 20 nodes. Here we present our findings for 20 seed nodes. Furthermore, only nodes with DEG ≥ 5 are enlisted. For each member in RND a biased sampling is performed towards one of its N_α neighbors ($\alpha = 1,2,3$) i.e., the neighbor with the respective highest centrality. Hence we obtain three seeds for each centrality, one composed of the N_1 neighbors of RND, a second from N_2 and a third one from the N_3 neighborhood. For example DEG-N1 replaces each node in RND with the highest DEG node in N_1 respectively. A random approach that selects random nodes from N_1, N_2 and N_3 is also employed for a baseline comparison.

9.3.2 Accelerating the Spreading Process

Initially RND will be evaluated for its spreading capabilities, i.e., power of influence, and then replaced and compared with the influence potential of its relatively close neighbors (N_1, N_2 and N_3). Specifically the SIR process will deploy as cascade initiators – the set of nodes initially in state I – the nodes of RND. The spreading power (SP) for the cascade initiators will be defined by the number of nodes in the R state at the end of the SIR process. Similarly for SIS, SP will be measured as the number of nodes in I state when SIS reaches the steady (equilibrium) phase. Likewise, the set of nodes from DEG-N1, PCI-N1, etc., will be used as cascade initiators and measured for their spreading potential. We establish our work on the basis of the paradox example, with aim to identify the set of nodes – or rather the set of neighbors – that accelerate the spreading process more efficiently.

In order to obtain unbiased results, SP is averaged over 1000 iteration for all selected seed sets in both spreading models. To ensure that the illustrated results are not the product of a specific random seed, the final results are obtained (averaged) over 20 different RNDs.

9.3.3 Blocking the Outspread of Misinformation

Additionally, we propose a baseline approach for mitigating the outspread of misinformation within an online social platform, e.g., Facebook, Twitter or LinkedIn. Specifically we envision a notification system that informs users for malicious ”data” traversing the network much like the weather alert system of Facebook that informs users for potential harsh weather conditions at their registered location. Hence similar to a weather notification, e.g., *”Good morning Pavlos, stay dry today in Volos. Rain is forecast”* we visualize the following *”Good morning Pavlos, be careful on post X, your friends have marked it as potentially fraud”*. Similar approaches have been deployed at the network of LinkedIn for recommendation systems based on a node’s ego-network [100]. The proposed mechanism for *Blocking the Outspread of Misinformation in sociAl Networks* (BOMAN) will deploy a set of nodes as ”guards” to counter potentially malicious data traversing the network. The message will appear for a user-node, if within his ego-network a guard exists. Guard nodes will be deployed as instructed by the set of nodes in e.g., PCI-N2, PR-N2, etc. Hence in a similar fashion we search for the set of neighbors that can more efficiently block the outspread of ”undesired” data. We believe that BOMAN will discourage node-users in believing a post when such notifications appear. We model this ”disbelief”, as a decrement in λ when a guard node exists in a node’s immediate neighbors. In order to evaluate the efficiency of the proposed mechanism, a random set of ”ill intentioned” nodes (of equal cardinality to the set of guardians) will be selected to initiate the ”deceitful” spreading process. Thus, an initially randomly selected set of guardian nodes will be evaluated for it’s blocking capabilities, and then replaced and compared with that of its close neighbors.

Likewise, to obtain unbiased results we utilize 20 random sets of ”ill intentioned” nodes, and for each such set 20 RND sets of guardians. Finally, each spreading process is repeated for 1000 iterations to obtain the final SP.

9.3.3.1 Spreading Evaluation

Figures 9.4 and 9.5 depict the impact of selecting ”central” neighbors within N_1, N_2 or N_3 of RND with respect to the spreading models for the Email-Enron and Brightkite networks. The y-axis depicts the spreading power (SP) for the selected seed sets whereas the x-axis shows the respective steps of propagation for SIR and SIS. Reminisce that the two spreading models stop at different conditions; SIR stops when there are no nodes left in the infectious state, whereas SIS finishes when a relatively fixed number of nodes remains infected. For our first observation related to the spreading models, it is straightforward that the cardinality of the set of nodes that remain infected in the equilibrium phase of SIS, will be lower when compared to the cardinality of the recovered nodes in SIR (see Supplementary Figures C.12 and C.19). Regarding the biased sampling performed towards the highest indexed nodes based on a centrality measure for SIR, e.g., PCI in either N_1, N_2 or N_3, we observe a significant increase in the number of influenced nodes with respect to RND or the random selection from any neighborhood. This observation can be explained by the paradox example, which as illustrated in Table 9.1 holds for majority of the network nodes, for all centralities and all evaluated networks. For instance the centrality paradox for CORE or PR, holds for more that 90% of the network nodes in Email-Enron or Brightkite. Hence, there exists strong possibility that by selecting a close neighbor of RND, a node possessing "richer" topological characteristics will emerge and thus potentially trigger a stronger spreading process, i.e., influence a larger subset of the network nodes. Nonetheless there are cases as illustrated in Figure 9.4 for the Email-Enron network, where for example RND-N_1 or N_2 coincide with DEG-N_1 near the end of the propagation. For the SIR model we attribute these occasions to network topology. This phenomenon is more evident for the SIS spreading model (Figure 9.5 bottom row), where the performance of the RND methodologies and especially of RND-N_1 and RND-N_2 is significantly enhanced, that is, the spreading power of RND-N_1 or RND-N_2 is closer (or coincides) to that of DEG, PCI or CORE with respect to the illustrated results for the SIR propagation at the later SIS steps. This observation holds for all evaluated networks (see Supplementary Figures C.19 to C.25). It can be explained by the nature of the SIS model −the exchange of node states from susceptible to infected and vice versa per spreading step− and the fact that the selected nodes are relatively close neighbors. The cascade initiators at the consensus will reach the same neighboring nodes that will preserve the "infection" in the network and hence sustain a relatively fixed number of nodes in I state as the spreading steps unfold.

Next, we will discuss the impact of selecting cascade initiators of the same centrality, but at different hop distance from RND. The most significant differences can be found at the early spreading steps for both SIR and SIS. Specifically it can be observed that for DEG, selecting N_3 neighboring nodes yields for almost all networks the largest SP, closely followed by DEG-N_2 with the exception of CA-GrQc network where N_2 takes the lead (see Supplementary Figure C.12). For instance, when focusing on lets say the 4^{th} spreading step of SIR we observe either a relatively small increase in N_3 with respect to N_1, e.g., less than 10% as in Hamsterster network or a vast increase in the number of influenced nodes as illustrated for the Brightkite or the CA-Astroph networks of about 40%. Similar conclusion can be drawn for the SIS spreading model, where likewise focusing on the 4^{th} SIS step in Figure 9.5 (left column), we observe an increment in SP for DEG-N_3 when compared to N_1 of about 15% for Email-Enron and about 45% for the Brightkite network. A similar performance where N_3 and N_2 cascade initiators compete for the first place and N_1 for the third at the early propagation steps, is illustrated for BC, CC and PR in the majority of the evaluated networks. CORE and ONION have similar performance, i.e., the obtained ranking between CORE-N_1, CORE-N_2 and CORE-N_3 for all evaluated networks

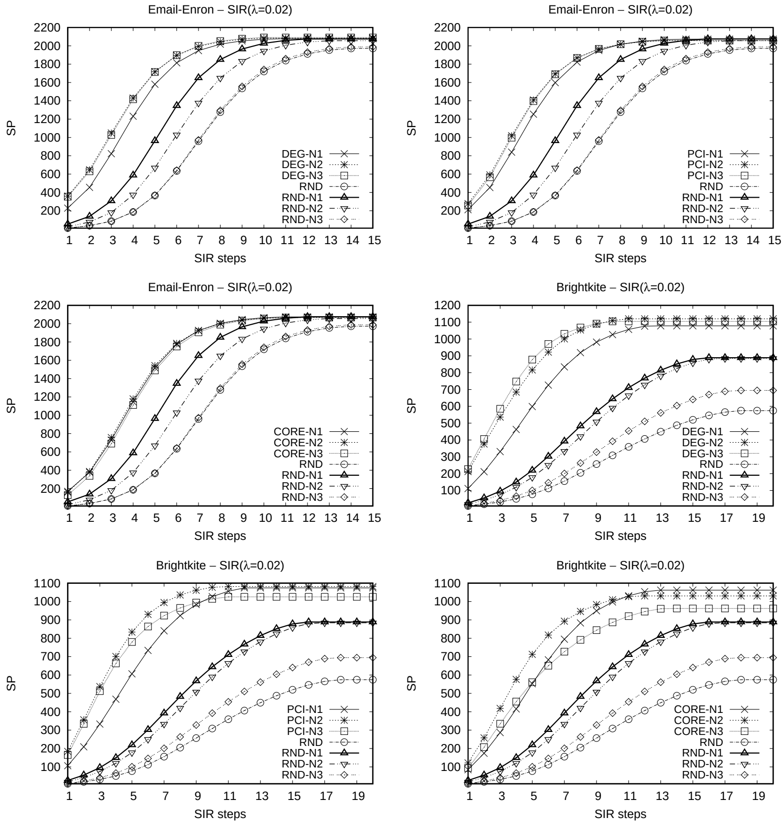


Figure 9.4: Influence maximization under the SIR spreading model for the *Email-Enron* and *Brightkite* networks for the DEG, PCI and CORE centralities.

is similar to the that of ONION (see Supplementary Figures C.14 and C.15). In other words when CORE-N_2 influences more network nodes than CORE-N_1 or CORE-N_3, the same applies for ONION respectively. PCI slightly deviates from the above observation. Focusing on RND, although "randomness" will play it's role in the observed results and the spreading potential of the respective initiators, it can be concluded that RND-N_1 outperforms RND in all evaluated networks for the SIR spreading model. On the other hand RND-N_2's performance varies with respect to N_1; coincides in CA-HepTh or Hamsterster networks, outperforms the competitor in CA-AstroPh and CA-CondMat, or has lower SP in Email-Enron or Facebook networks (see Supplementary Figure C.12). Finally, the performance of random initiators from N_3 is closer to RND, i.e., overall RND-N_3 influences a smaller portion of network nodes than RND-N_1 or N_2.

The illustrated results so far suggest that performing a biased selection among the neighboring nodes of an initially random selected set accelerates the spreading process. Our choice for the best policy in selecting important –by means of centrality– close neighbors would be DEG or PCI which depend on local knowledge of the network topology and thus combine efficiency and low computation cost. Among the neighboring sets, i.e., N_1, N_2 or N_3, we propose the N_2 set of nodes which showed similar performance to that of N_3 (but with less computation cost),

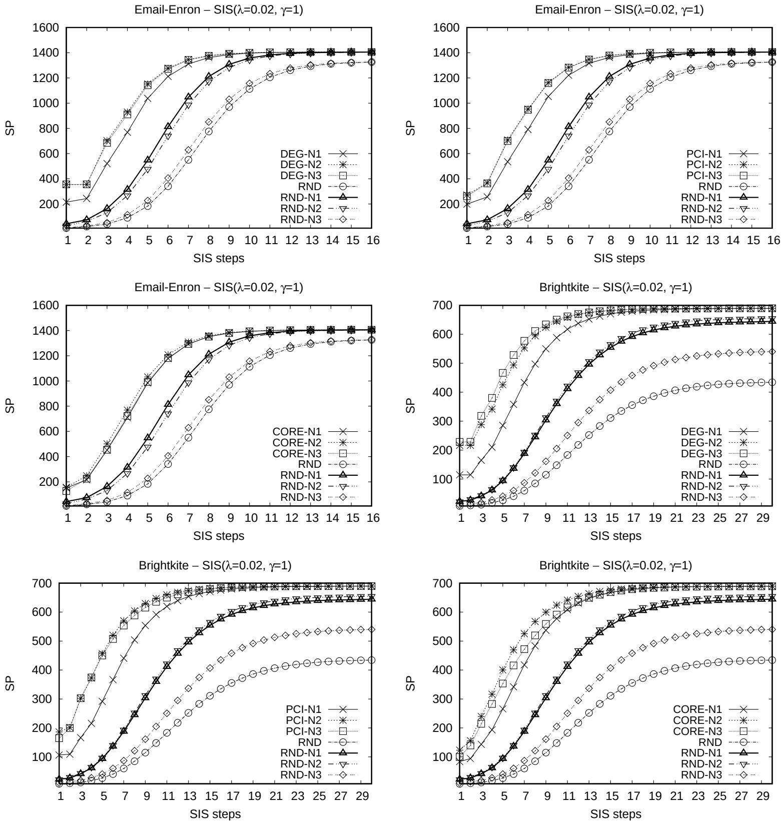


Figure 9.5: Influence maximization under the SIS spreading model for the *Email-Enron* and *Brightkite* networks for the DEG, PCI and CORE centralities.

and was found superior to the N_1 set of nodes. Overall, the paradox example favors the early propagation steps in terms of spreading power, since within a node's close proximity lie more connected nodes (DEG), nodes that reside in more dense neighborhoods (PCI), etc., that is, nodes that posses richer topological characteristics.

9.3.3.2 Blocking Evaluation

Figure 9.6 illustrates the results of BOMAN for the Email-Enron and Brightkite networks. The fraction of nodes influenced when no protection (NP) policy is active is illustrated for each network. The y-axis depicts the fraction of influenced nodes (SP) when BOMAN is active (with a corresponding guardian set) with respect to the unprotected outcome of the propagation. The dashed line in each plot shows the efficiency of BOMAN when RND is the set of guardian nodes. Finally, the "disbelief" factor will decrease λ by 0.2 when a guardian node exists within a node's ego-network. For our first observation, it can be concluded that the RND methodologies are the least effective strategies for protecting the network nodes. Focusing on the centrality metrics it can be observed that DEG-N_1 illustrated it's best performance in the Facebook network (see

Supplementary Figure C.11) where it performs equally to DEG-N_2, whereas in the remaining networks DEG-N_2 protected a larger subset of network nodes (see Figures 9.6, C.11). Similarly to DEG, (PCI, BC, CC, PR)-N_2 outperform their respective N_1 guardians in the majority of the illustrated results. This observation however, is less evident for CORE and ONION. Similar conclusions can be drawn when comparing the set of blockers from N_3 with that of N_1, i.e., in the majority of the illustrated results the outspread of misinformation is more efficiently hindered with N_3 guardians. When comparing the blocking capabilities between N_3 and N_2 guardians, DEG-N_3 is more efficient than DEG-N_2 in the majority of the evaluated networks. Although this observation holds also for PCI-N_3, CC-N_3, BC-N_3 or PR-N_3 respectively in several networks, e.g., for CA-CondMat or PGP (Figure C.11), the competitors showed also cases of very close performance or cases where, e.g., PCI-N_2 is a more effective blocking set than PCI-N_3 as shown in Figure 9.6.

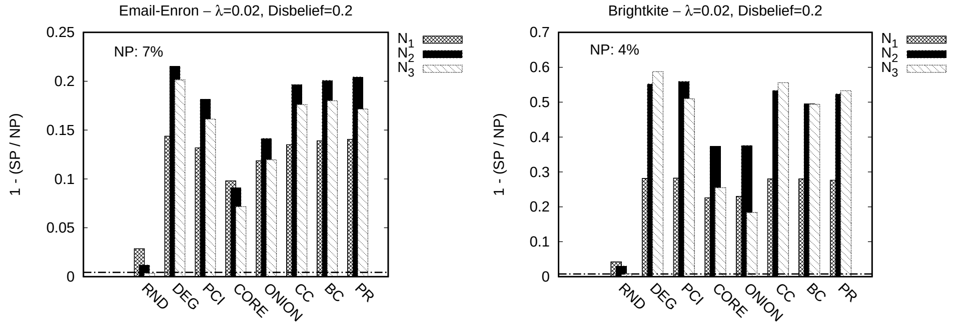


Figure 9.6: Blocking the outspread of misinformation for the *Email-Enron* and *Brightkite* networks under the SIR propagation model for all centralities.

Overall, in consensus with our conclusions from the previous section, we strongly suggest guardian nodes based on local knowledge of the network topology that combine efficiency and minimum computational cost. Although in terms of spreading potential selecting random nodes from N_1 or N_2 showed significant increase in SP with respect to RND, in terms of blocking capabilities, we find minimum or no improvement. BOMAN showed that it can be an effective mechanism to mitigate the outspread of ”undesired” data in networked populations by utilizing the paradox example, and hence mining more central nodes −more efficient blockers− within close proximity of an initially random selected set of nodes.

9.4 Discussion

Understanding dynamical processes in complex networks such as spreading processes to either accelerate propagation or hinder the outspread of undesired “things”, is of paramount importance that finds fertile ground in a plethora of applications. By considering different topological node characteristics (by means of centrality), we empirically found that the friendship “paradox” holds for a set of very popular centrality measures, not necessarily correlated to node degree, while it also holds for centrality measures that are not local, but recursive. Additionally, we show that neighbor superiority holds not only for the direct (one hop) neighbors of a node, but also for more distant (but close) neighbors. In other words we say: *your close friends have more friends than you (DEG)*, *your close friends reside in more dense neighborhoods than you (PCI)*, *your close friends are closer to the network core than you (CORE)*, *your close friends are part of more shortest paths than you (BC)*, *your close friends are closer to the remaining nodes than you (CC)*, *your close friends are pointed by more and more important nodes than you (PR)*.

Table 9.1: Fraction of nodes that the paradox holds at the individual level.

	CA-Astroph	CA-CondMat	CA-GrQc	CA-HepPh	CA-HepTh	PGP	Hamsterster	Facebook	Brightkite	Email-Enron
DEG	89.7 - 90.3 - 80.7	87.7 - 90.6 - 90.1	83.4 - 81.6 - 84.2	90 - 90.5 - 86.3	83.8 - 84.9 - 86.4	85.4 - 84.8 - 87.7	90.3 - 86.3 - 71.8	87.5 - 62.9 - 69.2	95 - 96.3 - 91.5	97.4 - 94.9 - 87.4
PCI	86.7 - 86.9 - 77.9	81.8 - 86.2 - 87.3	70.2 - 78.6 - 82.5	85.7 - 89.6 - 85.8	75.3 - 81.6 - 84.7	69.4 - 81.2 - 84.7	83.6 - 82 - 66.5	78.2 - 59.2 - 64.5	89.8 - 94.5 - 89.8	93.4 - 92.5 - 85.5
CORE	81.3 - 82.6 - 75.4	73 - 78.3 - 81.4	61.1 - 72.4 - 81.1	83.8 - 89 - 85.9	64.6 - 72.7 - 77.7	60.2 - 74.5 - 80.9	75.4 - 75.1 - 62.9	70.6 - 56.4 - 64	85.8 - 91.6 - 87.9	90.1 - 90.5 - 84.4
ONION	83.8 - 83.4 - 77	79.2 - 79.3 - 81.5	71.7 - 71.9 - 75.2	84.3 - 84.8 - 81.1	75.2 - 76.8 - 80.1	77.3 - 77.4 - 80.8	80.2 - 77.8 - 63.2	70 - 56.6 - 59.7	91.3 - 92.4 - 88.3	94 - 89.6 - 83.1
CC	82.4 - 87.1 - 76.5	80.5 - 88.2 - 86.8	78.9 - 83.7 - 85.3	82.2 - 89.2 - 82.5	77.3 - 84.7 - 86.4	80 - 87.4 - 87.7	82.8 - 83.3 - 63.2	91.5 - 85.5 - 72.7	88.5 - 93.8 - 87.4	93.4 - 90.9 - 78.8
BC	91.5 - 92.1 - 85.1	91.2 - 94.7 - 93.1	87.3 - 90.7 - 90.4	90.3 - 91.9 - 87.9	87.8 - 91 - 90.8	89.2 - 93.6 - 94.7	92.7 - 89.9 - 79.5	99.2 - 98.2 - 96.2	96.3 - 98.3 - 93.8	98.6 - 97.7 - 92.5
PR	87.2 - 88.8 - 80.1	86.3 - 88.5 - 88.2	82.6 - 80.4 - 83.3	86.2 - 87.5 - 82.7	82.6 - 83.4 - 84.9	84.9 - 83 - 85.8	89.9 - 86 - 72.5	89.6 - 59.9 - 70.1	93.9 - 96 - 91.1	97.4 - 94.9 - 87.6
-	-	-	-	-	-	-	-	-	-	-
SIR	70 - 76.4 - 76.5	67 - 75.4 - 77.8	62.5 - 72.6 - 85.5	66.9 - 81.6 - 86	63.9 - 74.4 - 77.3	68.2 - 76 - 85.8	87.7 - 85.6 - 70.2	69.9 - 70.3 - 67.7	70.6 - 86.9 - 88	76.6 - 83.7 - 83.8
SIS	84.6 - 88.6 - 80.5	65.2 - 74.2 - 76.5	35.8 - 55.5 - 73.7	76.4 - 89.6 - 87.7	64.8 - 74.3 - 77.2	53.2 - 68.9 - 80.1	87.5 - 85.4 - 69.8	73.5 - 62.1 - 70.5	61.9 - 84.4 - 86.9	93.2 - 92.6 - 85.7

Furthermore by differentiating our study from methodologies based on counting links incident upon the network nodes, we introduced the influential spreaders paradox by considering the spreading power of nodes (SP), i.e., influence, under the well established SIR and SIS spreading models. We thus embrace the probabilistic nature of the spreading paths embedded by the propagation models, and empirically show that indeed: *your close friends are more influential than you*. This conclusion applies strongly when SP quantifies the ability of the network nodes to infiltrate the networked environment, i.e., the SIR spreading model. For SIS, the subset of nodes that remain influenced throughout the spreading steps was found relatively steady and independent from the centrality measure (or neighborhood) used for selecting nodes from RND.

Complete (or accurate) knowledge of network topology on large-scale networks can be a very challenging and demanding task due to possible privacy constraints, in dynamically changing networks, when networks are processed in a streaming fashion, or for applications that need to meet time constraints, etc. Hence, mining more "central" nodes based on local information becomes increasingly important. We has show the effectiveness

Given the fact that the paradox holds for all evaluated centrality measures we empirically show that indeed selecting nodes within the near neighborhood of a randomly selected set reveals more central nodes. In terms of influence,

9.5 Materials and Methods

9.5.1 Data description

Ten real complex networks are studied in the paper. The network datasets are publicly available by the Stanford University [65] and by the University of Koblenz-Landau [8]. For those networks comprised of multiple connected components only the largest component is considered.

Table 9.2: Characteristics of examined complex networks. Apart from the number of nodes and edges, the table also depicts the epidemic threshold (ϵ), the average degree (k), and the type of the network.

Network	**Nodes**	**Edges**	ϵ (%)	λ	$\langle\mathbf{k}\rangle$	**D (90%)**	**A**	**Type**
CA-Astroph	17903	196972	1.5	0.02	22	14 (5)	0.201	Co-Authorship
CA-CondMat	21363	91286	4.4	0.05	8.5	14 (6.5)	0.127	Co-Authorship
Ca-HepPh	11204	117619	1	0.02	21	13 (5.8)	0.629	Co-Authorship
Ca-HepTh	8638	24806	7.7	0.08	5.7	17 (7.4)	0.239	Co-Authorship
Email-Enron	33696	180811	1	0.02	10.7	11 (4.8)	-0.116	Email
Brightkite	56739	212944	1.5	0.02	7.5	18 (5.7)	0.009	Social
Facebook	4039	88234	1	0.02	43.6	8 (4.7)	0.063	Social
Hamsterster	2000	16097	2.2	0.03	16	10 (4.8)	0.022	Social
PGP	10680	24316	5.3	0.06	4.5	24 (10)	0.238	Interaction
CA-GrQc	4158	13422	5.5	0.06	6.4	17 (7.6)	0.639	Co-Authorship

9.5.2 Individual and network level property

Suppose that the value of a measured feature (e.g., some centrality measure, or influential power) of a node i in a network with n nodes is v_i, and that the set of its neighbors (being either 1-hop neighbors thus $a = 1$, or 2-hop neighbors thus $a = 2$, or 3-hop neighbors thus $a = 3$) is denoted as $N_a(i)$. We would say that the paradox holds at the individual level of node i if the following condition holds true:

$$v_i < \frac{1}{|N_a(i)|} \sum_{\forall j \in N_a(i)} v_j. \tag{9.1}$$

On the other hand, we would say that the paradox holds at the network level if the following condition holds true:

$$\langle v \rangle = \frac{1}{n} \sum_{\forall i} v_i < \langle v \rangle_{nn} = \frac{\sum_{\forall i} |N_a(i)| \times v_i}{\sum_{\forall i} |N_a(i)|} \tag{9.2}$$

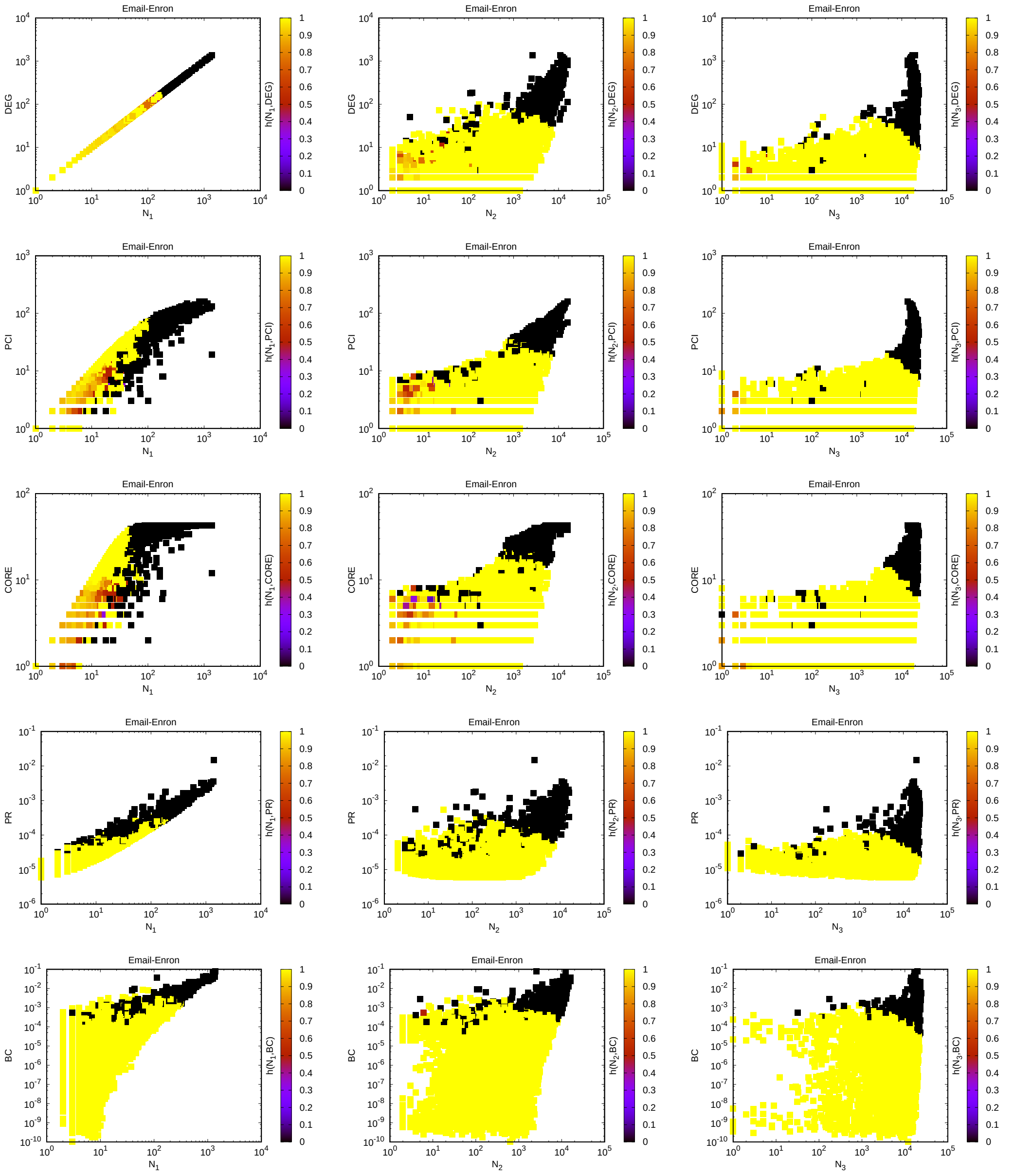


Figure 9.7: Evaluation of the centrality paradox at the individual level for the *Enron* network. Each line of plots corresponds to one centrality measure, namely DEG, PCI, CORE, PR and BC (the rest are given in the Supplement). Each column of plots corresponds to one neighborhood, namely the leftmost column is for 1-hop neighbors, the middle column of plots is about 2-hop neighbors, and the rightmost column of plots is about 3-hop neighbors. The x-axis in each plot depicts the size (in number of nodes) of the respective neighborhood, and the y-axis depicts centrality values. The heat values in the palette depict the centrality paradox holding probability. We observe that for a fixed neighborhood size, the centrality paradox holding probability decreases with increasing centrality value, for any centrality measure, and for all close neighborhoods. For some centralities, namely PR and BC this behavior is strictly 'binary', i.e., the centrality paradox either holds or not, no matter what the size of the neighborhood is. This binary behavior for all centralities is prevalent in N_2 and even more prevalent in N_3.

Part V

Implementation Issues on the Hadoop Environment

Chapter 10

Hadoop MapReduce performance on SSDs

Hadoop MapReduce performance on SSDs for analyzing social networks

10.1 Introduction

In this final chapter we investigate on the performance gain introduced in the Hadoop environment by utilizing solid state discs (SSDs) for analyzing computational intensive properties of social network, e.g., discovery of communities, spreading paths and connected components, etc. Modern social networks are comprised by millions of nodes and even billions of edges; therefore any algorithm for their analysis that relies on a single machine (centralized) - exploiting solely the machine's main memory and/or its disk - is eventually doomed to fail due to lack of resources. Thus, the digitization of the aforementioned relationships produces a vast amount of collected data, i.e., big data [55] requiring extreme processing power that only distributed computing can offer. However, developing a distributed solution is a challenging task because it must deal sometimes with sequential processes. Some analysis algorithms based on distributed solutions that can run only on a small cluster of machines are still insufficient, since modern OSNs are maintained by Internet giants such as Google, LinkedIn and Facebook who own huge datacenters

Related publication [**J2**]: Marios Bakratsas, Pavlos Basaras, Dimitrios Katsaros, Leandros Tassiulas. *Hadoop MapReduce performance on SSDs for analyzing social networks*, **Big Data Research (Elsevier)**, accepted, June, 2017.

Related publication [**C1**]: Marios Bakratsas, Pavlos Basaras, Dimitrios Katsaros, Leandros Tassiulas. *Hadoop MapReduce performance on SSDs: The case of complex network analysis tasks*, **Proceedings of the 2nd Neural Network Society International Conference on BigData (INNS BigData)**, chapter in Advances in Big Data, series in Advances in Intelligent Systems and Computing, vol. 529, pp. 111-119, Thessaloniki, Greece, October 23-25, 2016.

and operate clusters of several thousand machines. These clusters are usually programmed by data-parallel frameworks of the MapReduce type [179], a big data analytics platform.

The Hadoop [47] middleware was designed to solve problems where the "same, repeated processing" had to be applied to peta-scale volumes of data. Hadoop's initial design was based on magnetic disk's characteristics, enforcing sequential read and write operations introducing its own distributed file system (HDFS - Hadoop Distributed File System) with blocks of large size.

Recently with the advent of faster Solid State Drives (SSDs) research is emerging to test and possibly to exploit the potential of the new technologically advanced drive [14], [59], [60], [68]. The lack of seek overhead gives them a significant advantage with respect to Hard Disk Drives (HDDs) for workloads whose processing requires random access instead of sequential access. Even though the cost-per-capacity of SSDs is still high, their adoption could be widespread if their performance was solidly proved to be superior to that of HDDs. The world of databases has long time ago started [161] to assess the benefits of using SSDs in various points of the database architecture, but the Hadoop world has only recently [60], [68], [92], [104] started a similar investigation.

Providing a clear answer to the question of whether SSDs significantly outperform or offer increased performance in same cases compared to HDDs in the Hadoop environment is not straightforward, because the results of a system-analysis-based investigation are affected by the network speed and topology, by the cluster (size, architecture,...), and by the nature of the benchmarks used (MapReduce algorithms, input data). The efforts done so far to provide light to this question suffer either because the experimentation was executed on a virtualized cluster [92], or because their setup was affected by the underlying network [68], or because their benchmark algorithms and data were mostly read-oriented [60], [68], thus biasing the results in such a way that no clear answer and universally holding conclusions could be drawn.

This article attempts to start the investigation from a new basis and to provide a clear answer to the following basic question: Ignoring any network biases and storage media cost considerations, do SSDs provide improved performance over HDDs for real workloads that are not dominated by either reads or writes? In this context, our article makes the following contributions:

- It uses a different set of MapReduce jobs, i.e., complex network analysis tasks, which have radically different characteristics from the earlier used benchmarks.

- It isolates "external" dependencies, i.e., network, cost considerations.

- It shows that there exists at least one case where HDDs can deliver superior performance to SSDs, which has not been documented in any earlier study.

- It provides solid evidence that the MapReduce job's read/write behavior will eventually provide the answer of whether SSDs are preferable over HDDs, which is consistent with the conclusions reported in [117] where random writes in SSDs are the "killing" application pattern for SSDs (with respect to reads and sequential writes).

The rest of the article is organized as follows: In section 10.2 we present the related work, and in section 10.3 we briefly describe Hadoop's structure. In section 10.4, we provide information about the three algorithms that will be evaluated in the storage media. Section 10.5 contains the evaluation results, and finally, section 10.6 concludes the article.

This paper is based on an earlier look on this topic [9]. In particular, the main augmentation parts in the current paper are the following ones: section 10.2 has been expanded significantly including more related works; section 10.3 which gives a brief overview of Hadoop architecture; the whole section 10.4 which presents in details the examined algorithms is practically new material (only Table 10.1 appears in the conference version of the article); section 10.5.3.1 which evaluates the competing disks against an industry standard is new material; and finally, performance results presented in Figure 10.7 and Figure 10.8 along with the associated explanations are also new material.

10.2 Related work

Introducing and investigating the usage of SSDs in Hadoop clusters has been a hot issue of discussion very recently. The most relevant work to ours is included in the following articles [60], [68], [71], [92], [104]. The first effort [92] to study the impact of SSDs on Hadoop was on a virtualized cluster (multiple Hadoop nodes on a single physical machine) and showed up to three times improved performance of SSDs versus HDDs. However, it remains unclear whether the conclusions still hold in non-virtualized environments. The work in [68] compared Hadoop performance on SSDs and HDDs on hardware with non-uniform bandwidth and cost using the Terasort benchmark. The major finding is that SSDs can accelerate the shuffle phase of MapReduce. However, this work is confined by the very limited type of application/workload used to make the investigation and the intervention of data transfers across the network. Cloudera's employees in [60], using a set of same-rack-mounted machines (not reporting how many of them), focus on measuring the relative performance of SSDs and HDDs for equal-bandwidth storage media. The MapReduce jobs they used are either read-heavy (Teravalidate, Teraread, WordCount) or network-heavy (Teragen, HDFS data write), and the Terasort which is read/write/shuffle "neutral". Thus, neither the processing pattern is mixed nor the network effects are neutral. Their findings showed SSD has higher performance compared to HDD, but the benefits vary depending on the MapReduce job involved, which is exactly where the present study aims at.

The analysis performed in [71] using Intel's HiBench benchmark [125], [137] concluded that "...the performance of SSD and HDD is nearly the same", which contradicts all previously mentioned works. A study of both pure (only with HDDs or only with SSDs) and hybrid systems (combined SSDs and HDDs) is reported in [104] using a five node cluster and the HiBench benchmark. Differently from the present work, in that work, the authors investigated the impact of HDFS's block size, memory buffers, and input data volume on execution time showing that when the input data set size and/or the block size increases, then the performance gap between a pure SSD system with a pure HDD system widens in favor of the SSD system. Moreover, for hybrid systems, the work showed that more SSDs result in better performance. These conclusions are again expected since voluminous data imply increased network usage among nodes.

Earlier work [114], [145] studied the impact of interconnection on Hadoop performance in SSDs identifying bandwidth as a potential bottleneck. The increase of bandwidth by using high-performance interconnects benefits HDFS performance on both disk types, but especially SSDs. Both conclusions are expected since a lot of data transfer takes place among nodes in map-shuffle-reduce operations. Less related to our study, [26] proposes a performance model using queuing network to simulate the execution time of MapReduce and thus come up with a cost-performance model for SSDs and HDDs in Hadoop, and [19], [40] explore how to optimize a Hadoop MapReduce framework with SSDs in terms of performance, and/or cost/energy.

Finally, some works propose extensions to Hadoop with SSDs. For instance, [59] proposes extensions to enable clusters of reconfigurable active SSDs to process streaming data from SSDs using FPGAs. VENU [63] is a proposal for an extension to Hadoop that will use SSDs as a cache for the slower HDDs not for all data, but only for those that are expected to benefit from the use of SSDs. This work still leaves open the question about how to tell which applications are going to benefit from the performance characteristics of SSDs. Remotely related to our work is the discussion about the introduction of SSDs in database systems, e.g., [161].

10.3 Hadoop structure

Hadoop is an open source framework, written in the Java programming language which allows for processing large data sets in a parallel/distributed computing environment. HDFS and MapReduce (MR) are the two core components of Apache Hadoop.

HDFS is Hadoop's distributed file system that provides high-throughput access to data, high-availability and fault tolerance. Data are saved as large blocks (default size 128MB) making it suitable for applications that have huge data sets. It creates replicas of each block and distributes them among the nodes of the cluster.

MapReduce is a software framework that allows to write applications and execute them upon a cluster comprised by a few machines to several thousand commodity machines. It takes care of all cluster maintenance tasks and job scheduling operations and allows the programmer to focus on programming the logic of the application. Submitting a MapReduce job to the master node, results in splitting the input "file" to several chunks (block sized) that are processed by Map and Reduce tasks at parallel. Due to block replication of HDFS, tasks are scheduled to run on nodes where the required chunks of data already exist, minimizing unnecessary transfer of these data.

The key functions to be implemented are Map and Reduce. The MapReduce framework operates on (key,value) pairs. Each Map task processes an input split (block) generating intermediate data of (key,value) format. Then, they are sorted and partitioned by key, so later at Reduce phase, pairs of the same key will be aggregated to the same reducer for further processing. The flow of data is depicted in Figure 10.1. Here lays Hadoop's main advantage. Partitions from different nodes with the same key are transferred (shuffle phase) to a single node and then merged (sort phase) and get ready to be fed to the reduce task. The output of Reduce tasks is of format (key, value) as well.

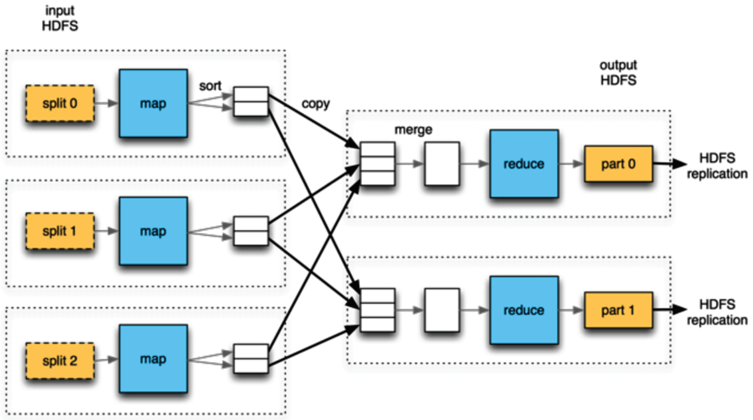


Figure 10.1: Overview of Map/Reduce and Hadoop (from [47]).

10.4 Investigated algorithms

Complex network analysis comprises a large set of diverse tasks (algorithms for finding communities, centralities, network growth models, resilience to attacks, epidemics, etc) that cannot be enumerated here, and whose particular form depends on the field of study (technology, biology, sociometry, medicine) and also on the particular application that the "human miner" is interested in. Apparently, not all these tasks accept distributed solutions (at least, efficient ones) in the form of MapReduce algorithms, but there is already a significant body of works that developed MapReduce algorithms for solving problems such as triangle enumeration [62], k-shell computation [69], k-means clustering [157], neural networks [76], etc.

Therefore, among all these problems and their associated MapReduce solutions, we had to select some of them based on a) their usefulness in complex network analysis tasks, b) in their suitability to the MapReduce programming paradigm, c) the availability of their implementations (free/open code) for purposes of reproducibility of measurements, and d) complexity in terms of multiple rounds of map-reduce operations. Based on these criteria, we selected three problems/algorithms for running our experimentations. The first algorithm deals with a very simple problem which is at the same time a fundamental operation in Facebook , that of finding mutual friends. The second algorithm deals with a network-wide path-based analysis for finding connected components which finds applications in reachability queries, techniques for testing network robustness and resilience to attacks, epidemics, etc. The third algorithm is about counting triangles which is a fundamental operation for higher level tasks such as calculating the clustering coefficient, or executing community finding algorithms based on clique percolation concepts [176]. We wanted to have problems that deal with both the local and global structure of the network. Table 10.1 summarizes the "identity" of the examined tasks.

Primitive	**Type of analysis**	**Extent**
Mutual friends	Neighbor-based	Local network (neighborhood) properties Recommendation queries
Connected components	Path-based	Large-scale network properties, Reachability queries, Resilience queries
Triangle counting	Mixed (extended neighborhood & paths)	Large-scale network properties, Clustering/communities finding queries

Table 10.1: Characterization of problems/algorithms examined.

We need to emphasize that it is not the purpose of this article to develop a benchmark suite of algorithms and input data for MapReduce, even though we clearly recognize this need and call for the development of a really generic and representative benchmark; current efforts in this topic (like the Hibench [125], [137]) are in a rather infantile age and their tasks (wordCount, k-means clustering, Bayesian classification, PageRank, etc) are mostly appropriate for information retrieval or basic, traditional data mining tasks. So, our benchmark includes representative (in the notion described above) MapReduce jobs to cover common IO patterns expected to be seen in complex network analysis. We deferred a more advanced method for measuring the performance for multi-job workload such as the one described in [124], because the standalone, one-job-at-the-time method allows for the examination of interaction between MapReduce and storage media without the interventions of job scheduling and task placement algorithms.

We aim at showing that the conclusions about the relative performance of SSDs versus HDDs are strongly depended on the features of the algorithms examined, which has largely been neglected in earlier relative studies [60], [68], [92], and based on these features we draw some conclusions on the relative benefits of SSDs. For purposes of the article's self-completeness, we present in the following three sections the selected algorithms and a brief explanation of their operation.

10.4.1 Mutual friends

A common feature of various social networks is providing information of the existence of mutual friends once visiting some other user's profile page. A simple algorithm was implemented for the calculation of mutual friends. The necessary condition is that this pair of users are already friends (connected) with each other. Pseudocode for the MapReduce algorithm is given in Figure 10.2.

```
%1st MR job - CalculateAdjacencyList:
ON MAP DO:
 for each KV pair do:
    K<-source_node
    V<-destination_node
    context.write (K,V)
    context.write (V,K)
ON REDUCE DO:
 for each K[V] pair do:
    ego_user<-get(k)
    for each v in V
        add v to nodes_list
    sort the nodes_list
    for each node_id in nodes_list
        append node_id to friendlist
    context.write (ego_user,friendlist)

%2nd MR job - Creating triples:
ON MAP DO:
 for each KV pair do:
    K<-ego_user
    V<-friendlist
    for each friend in friendlist
        for each other_friend in friendlist
            if ego_user<friend then
                context.write (ego_user-friend:other_friend , NULL)
            else
                context.write (friend-ego_user:other_friend , NULL)
ON REDUCE DO:
 for each KV pair do:
    if |V|==2 then
        context.write (triple,NULL)

%3rd MR job
ON MAP DO:
 for each KV pair do:
    pair_and_mutual=K.split(":")
        pair=pair_and_mutual(0)
        mutual=pair_and_mutual(1)
        context.write (pair,mutual)

ON REDUCE DO:
 for each KV pair do:
    pair<-get(K)
    for each v in V
        v<-mutual
        mutuals_list.add(mutual)
 context.write (pair,mutuals_list)
```

Figure 10.2: MapReduce pseudo-code for finding mutual friends.

The basic idea behind the algorithm is for every user (i.e., node) and his friend-list (i.e., adjacency list) to create all possible triples consisting of:

- The owner of the friend-list,
- A user of the friend-list who will make a pair with the owner, and
- Another user of the friend-list who will be the candidate mutual friend.

The same work is performed for each and every user and his friend-list. Eventually, if two exact triples are spotted, then the candidate is classified as a mutual friend for the specified pair. For the implementation three MR jobs are required:

1. Calculation of the adjacency list (friend-list). The input file is a graph containing all the ties among the nodes. Each node is a number unique for each user. All used social

network datasets, were un-weighted, undirected graphs. Each line consists of a source node and destination node. Duplicate relationships aren't present in the original files. On the contrary, such supplementary information is necessary for the creation of adjacency lists, thus created by the Map function. Reduce function produces lines of every node and its adjacency list.

2. Creation of all available triples according to the basic concept that was mentioned previously. The Mapper output creates all available triples as key. Value is set to NULL. At Reducer, for a specific Key aggregating two NULL values, confirms the existence of a mutual friend.

3. Creation of the lists of mutual friends. At the Mapper, from each triple the pair is extracted as Key and their mutual as Value. The Reducer completes the creation of mutual friends list for every pair.

10.4.2 Connected components

Another very useful and primitive process of complex network analysis is the detection of connected components i.e., clusters of nodes where every node of the cluster can be eventually be accessed by any other node of the cluster following a path of arbitrary number of hops. This task finds applications in reachability analysis, in epidemics, i.e., once isolated users or groups are found, the spread of a contagion can be stopped, etc.

For this task, the implementation by Thomas Jungblut [6] of an iterative algorithm based on message passing technique is used (see Figure 10.3).

At the first iteration, the algorithm maps every first element as key and its adjacency list in vertex form as a pointsTo tree. Also, it maps each edge of the tree in vertex form. At reduce, the algorithm marks all vertexes having a pointsTo tree as activated. It sets the smallest element of this list (comparing to the key as well), as vertex's minimal. Then, it writes key and vertex in context. At next iterations, map writes each key and vertex as it is. Also for every activated vertex, it loops through the pointsTo tree and writes a message (vertex with empty tree) with the (for this vertex) minimal vertex to every edge of the tree. At reduce, it merges messages with the related vertex and if a new minimum is found then activates the vertex. The updated counter gets incremented. Otherwise deactivates the vertex. Iterations continue till no vertex gets updated.

10.4.3 Counting triangles

Counting the number of triangles in a graph is a fundamental problem with various applications especially in social network analysis. For example, the clustering coefficient is frequently quoted as an important index for measuring the concentration of clusters in graphs respectively its tendency to decompose into communities.

```
%1st MR job
ON MAP DO:
 for each line (adjacency list)
    realkey<-first edge of adjacency list
    vertex<-all other edges sorted, plus minimal
    context.write (realkey, vertex)
 for all edges in vertex
    context.write (edge, new empty vertex with edge as minimal)

ON REDUCE DO:
 for each KV pair do:
    if V is not message then
       realVertex<-edges of V
       activate realVertex
       increment UPDATED counter
       context.write(key,realVertex)

%2nd MR job
ON MAP DO:
 for each KV pair do:
    context.write (K,V)
    if V is activated then
        for all edges in V
           if edge != minimal of V
               newVertex<-null edges
               newVertex<-minimal of V
               context.write (edge, newVertex)

ON REDUCE DO:
 for each K[V] pair do:
    for every v in V
        if v is not message then
            realVertex<-v
        else
            track newMinimal among messages v in V
    if realVertex.minimal > newMinimal then
        update realVertex with the lower newMinimal
        activate the realVertex
        increment UPDATED counter
    else
        deactivate the realVertex
    context.write(key, realVertex)
```

Figure 10.3: MapReduce pseudo-code for finding connected components.

We used the implementation by Walkauskas [7] (pseudo-code in Figure 10.4) which includes three MapReduce jobs:

- A triangle exists when a vertex has two adjacent vertexes that are also adjacent to each other. The first job constructs all of the triads in the graph. A triad is formed by a pair of edges sharing a vertex, called its apex. Original edges are written, as well. The above are written as keys with the value of 1 or 0 respectively to distinguish triads from original edges.

- The second MapReduce job maps previous input line, and the Reducer aggregates the triads with the edges for a specific triple. In order for a triangle to exist, there should be at least one candidate triad and the edge connecting the apex. The reducer eventually writes sum to context as “0, sum”.

- The third MapReduce job aggregates the number of triangles that was found from previous job for all chunks.

```
%1st MR job - TriadConstruction:
ON MAP DO:
 for each KV pair do:
    if K < V write to context

ON REDUCE DO:
 for each K[V] pair do:
    for each v in V
        save v in Array
        context.write (Kv, "zero")
    sort the Array
    for each v of sorted Array
        for each v' following v in the Array
            context.write (vv', "one")

%2nd MR job - TriadConstruction:
ON MAP DO:
 for each KV pair do:
    K<-source_node
    V<-destination_node
    context.write (K,V)

ON REDUCE DO:
 for each K[V] pair do:
    sum all v values in V
    compare the sum to the #v in V
    if not equal
        increase #triangles found by sum
    context.write(zero, count)

%3rd MR job - AggregateTriangles:
ON MAP DO:
 for each KV pair do:
    K<-source_node
    V<-destination_node
    context.write (K,V)

ON REDUCE DO:
 for each K[V] pair (only one pair with "zero" key) do:
    sum all v in V
 context.write (sum, null)
```

Figure 10.4: MapReduce pseudo-code for triangle counting.

We see that all three algorithms are executed in two or more pairs of 'maps' and 'reduces' which is a desired complexity for our measurements in terms of read and write operations.

10.5 Experimental environment and results

In this section we describe the system's setup and then we provide the obtained results for each one of the three algorithms presented earlier.

10.5.1 System setup

A commodity computer (Table 10.2) was used for the experiments. Three storage media were used (Table II) with capacities similar to that used in [68]. On each of the three drives (one HDD and two SSDs) a separate and identical installation of the required software (Table 10.3) was used. We emphasize at this point that since we need to factor out the network effects, we used single machine installations. Three different incremental setting setups were used: a) with default settings, allowing 6 parallel maps, b) with modified containers allowing 3 parallel maps, and c) with custom settings (Table 10.4). In all these setups, speculative execution was disabled and no early shuffling was permitted. We admit the a shortcoming of our study is the fact that we do not have a clear view of the types of storage devices used in the datacenters of the Internet giants (Google, Facebook), but still we are confident that the relative performance of the devices used will support our arguments. Power saving options and boosting technologies like Turbo-boost and IEST were disabled through BIOS to minimize unexpected fluctuations among executions.

CPU	Intel i5 4670 3.4Ghz (non HT)
RAM	8gb 1600mhz DDR3 (1333mhz with disabled XMP)
Disk 1 (HDD)	Western Digital Blue WD10EZEX 1TB
Disk 2 (SSD1)	Samsung 840 EVO 120GB
Disk 3 (SSD2)	Crucial MX100 512GB

Table 10.2: Computer specifications.

OS	Ubuntu 14.04 LTS 64bit
Java SDK	Oracle Java 1.8.0_25 (8u25)
Hadoop version	Hadoop 2.5.2 (pre-built 32-bit i386-Linux native Hadoop library)
Monitoring tools	Collectl V3.6.9-1

Table 10.3: Installed software.

mapreduce.reduce.shuffle.parallel.copies	5 – 50
mapreduce.task.io.sort.factor	10 – 100
mapreduce.map.sort.spill.percent	0.80 – 0.90
io.file.buffer.size	4KB – 64KB

Table 10.4: Custom settings.

10.5.2 Input data and performance measures

For the evaluation of the two disk types a sample of real data was required. Recall that earlier efforts e.g., [68] used dummy data files that were read and some primitive statistics were written out. Social networks is a representative sub-genre of complex networks. Thus up to ten real social network graphs were used (Table 10.5). They were retrieved from https://snap.stanford.edu/ and http://konect.uni-koblenz.de/. The number of nodes and edges vary from a few thousands to a few millions. Thus, we used networks that vary up to two orders of magnitude in their size (number of nodes and/or edges).

The evaluation will take place along two dimensions. The first one is similar to that in [68] using TestDFSIO and the second one is the complex network analysis-oriented that is the focus of this article. We have performed up to five experiments for each of the “Mutual Friends” and

	Social network	**# nodes**	**# edges**
1	Brightkite location based online social network	58,228	214,078
2	Gowalla location based online social network	196,591	950,327
3	Amazon product co-purchasing network	334,863	925,872
4	DBLP collaboration network	317,080	1,049,866
5	YouTube online social network	1,134,890	2,987,624
6	YouTube (ver. 2) online social network	3,223,589	9,375,374
7	Flickr	1,715,255	15,550,782
8	LiveJournal online social network	3,997,962	34,681,189
9	LiveJournal (ver. 2) online social network	5,204,176	49,174,620
10	Orkut online social network	3,072,441	117,185,083

Table 10.5: Social networks used for evaluation.

"Counting Triangles" algorithms and up to ten experiments for the "Connected Components", one for each dataset shown at Table 10.5. The latter algorithm acquired less disk space during execution allowing us to evaluate it with larger datasets. The two SSDs were of different size disallowing the execution of some datasets. The most important measures we captured were the Map and Reduce execution times, as also Sort (merge) and Shuffle phase All measured times are in seconds, unless otherwise stated. The aforementioned measures would indicate practical performance differentiations between the two disk types. One common side effect is "cache hits" from previous executions that was also experienced in [68]. In order to give each experiment an equal environment to eliminate any possible interaction effects from previous executions, Hadoop was halted and page cache was flushed, after each experiment. Before each test HDFS was re-formatted.

10.5.3 Results

10.5.3.1 TestDFSIO

We begin with the HDFS throughput measurement. Test Distributed File System (TestDFSIO) is an industry-standard benchmark which distributes map tasks that read/write complete dummy files on nodes; each map task reads the complete file and writes some statistics. Reduce tasks simply gather these statistics for output.

The write throughput performance is presented in Figure 10.5. We observe that for writing sequential files, with the increase of filesize, SSD1's performance is decreasing, falling behind the HDD. Contrariwise, the SSD2 appears much faster with stable throughput. The 120GB Evo, features a second level TurboWrite Cache (TWC). This 3GB block of high speed SLC memory allows the EVO to write data (nominally) at 370 MB/s, nearly double its normal rate. However, when the TWC is full or can not be used effectively, write speeds drop by around 50%, and this is the pattern that we observe in the plot.

The sequential read performance of the competitors is presented in Figure 10.6. As expected,

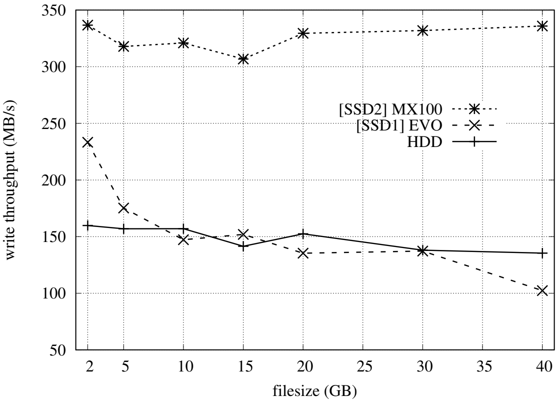


Figure 10.5: Comparing TestDFSIO write throughput for 3 disks.

both SSDs's sequential read throughput is outstanding. Moreover, both SSDs attain a read performance close to that given by their specifications, namely 540MB/s for SSD1 and 550MB/s for SSD2, and it is practically stable and independent on file size. On the other hand, the magnetic disk again demonstrates stable performance, although noticeably slower than that of the SSDs.

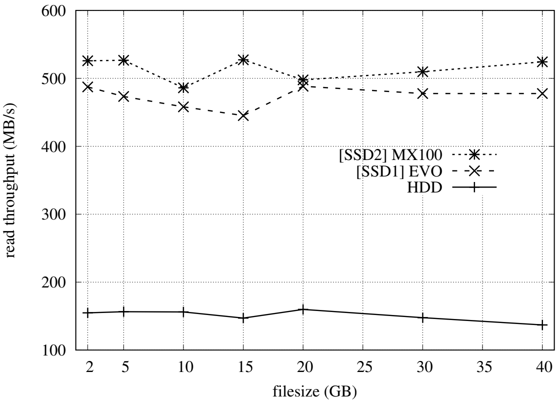


Figure 10.6: Comparing TestDFSIO read throughput for 3 disks.

10.5.3.2 Results on finding mutual friends

The complexity of this algorithm is exponential due to the mapper of the 2nd MapReduce job ("creating triple" - as described at section 10.4.1) where for each user and his friend-list every possible triple is formed (double "for" used). Thus, the 2nd MapReduce job is the most resource-intensive of the three jobs, rendering it a good inspection point for our measures (see Table 10.6), whereas the 1st and 3rd MapReduce jobs were fast-executed and almost identical for all disks. For Amazon, Brightkite and DBLP, the three disks performed almost equally. Remarkably, in comparison with both SSD drives, the magnetic disk gives *competitive (and slightly better) execution times for reduce phase for bigger datasets*, whereas HDD performs lower for map phase. The SSD2 displays superior performance at shuffling.

	Avg Map			Avg Shuffle			Avg Merge			Avg Reduce		
	HDD	SSD1	SSD2	HDD	SSD1	SSD2	HDD	SSD1	SSD2	HDD	SSD1	SSD2
Brightkite	52	52	52	1	1	1	0	0	0	11	10	10
Amazon	36	35	35	2	1	1	0	0	0	8	7	8
Gowalla	1780	1752	1593	120	103	42	0	0	0	178	195	194
DBLP	90	89	89	5	2	3	0	0	0	16	17	17
YouTube	11197	-	9708	812	-	258	0	-	0	916	-	984

Table 10.6: Average times for each phase for 2nd job (creating triples) of “mutual friends” algorithm.

10.5.3.3 Results on counting triangles

Here, the SSDs outperform the HDD for all the datasets that were tested. At “forming the triads” job, HDD appeared competitive behavior at map and reduce phases (Table 10.7). The “counting the triangles” job demonstrated greater variations in execution times. With small datasets the performance differentiations between the two disk types are small (Table 10.8). But with larger ones (like YouTube dataset), SSDs capabilities become evident for shuffle and merge (sort) phases.

	Avg Map		Avg Shuffle		Avg Merge		Avg Reduce	
	HDD	SSD2	HDD	SSD2	HDD	SSD2	HDD	SSD2
Gowalla	2	2	1	1	0	0	142	140
YouTube	6	6	1	1	0	0	706	694
Flickr	13	13	1	1	0	0	5053	5125

Table 10.7: Average times for each phase for 1st job (forming triads) of “counting triangles” algorithm.

	Avg Map			Avg Shuffle			Avg Merge			Avg Reduce		
	HDD	SSD1	SSD2	HDD	SSD1	SSD2	HDD	SSD1	SSD2	HDD	SSD1	SSD2
Brightkite	18	18	18	1	1	1	0	0	0	4	4	3
Amazon	9	9	9	1	1	1	0	0	0	2	2	2
Gowalla	38	39	38	52	62	21	79	86	70	106	106	110
DBLP	14	14	14	1	1	1	0	0	0	7	5	5
YouTube	42	-	41	655	-	141	820	-	668	689	-	551

Table 10.8: Average times for each phase for 2nd job (counting triangles) of “counting triangles” algorithm.

For the 1st MR job (creating triads), map, shuffle and merge phases finished quite fast and with almost zero differentiations among disks. Reduce phase lasted significantly longer with both disks performing equally (Table 10.6). With containers settings, the biggest dataset of Flickr gets significant improvement for both disk types (Table 10.9). No further improvement achieved with custom settings.

	Avg Map		Avg Shuffle		Avg Merge		Avg Reduce	
	HDD	SSD2	HDD	SSD2	HDD	SSD2	HDD	SSD2
Gowalla	2	2	1	1	0	0	141	138
YouTube	6	6	1	1	1	1	697	707
Flickr	13	13	1	1	6	6	4163	4140

Table 10.9: Average times for each phase for 1st job (create triads) of "counting triangles" algorithm, with changed container's settings.

To optimize performance, increasing the following settings provided best results for the magnetic disk, compared to "containers" settings:

a) The number of streams to merge at once while sorting files.

We see (Table 10.10 and Table 10.11) that it minimizes merge time for both disk types, but it improves the shuffling time of the HDD only. Even though both disks are able to reap benefits from this settings, HDD gains the most.

[HDD] just containers and varying io.sort.factor					
	Elapsed	Avg Map	Avg Shuffle	Avg Merge	Avg Reduce
io.sort.factor:10	52mins, 43sec	25	565	596	720
io.sort.factor:100	40mins, 26sec	25	471	14	667

Table 10.10: Performance difference for YouTube dataset at "Counting Triangles", increasing sort factor, for HDD.

[SSD2] just containers and varying io.sort.factor					
	Elapsed	Avg Map	Avg Shuffle	Avg Merge	Avg Reduce
io.sort.factor:10	41mins, 08sec	25	359	339	535
io.sort.factor:100	35mins, 15sec	25	371	16	497

Table 10.11: Performance difference for YouTube dataset at "Counting Triangles", increasing sort factor, for SSD2.

b) The buffer size for I/O (read/write) operations.

Examining the impact of this change (Table 10.12 and Table 10.13), we observe that only the HDD is able to exploit efficiently, whereas its impact on SSD2 is mixed and insignificant.

[HDD] just containers and io.file.buffer.size					
	Elapsed	Avg Map	Avg Shuffle	Avg Merge	Avg Reduce
io.file.buffer.size: 4KB	52mins, 43sec	25	565	596	720
io.file.buffer.size: 128KB	46mins, 44sec	25	445	470	619

Table 10.12: Performance difference for YouTube dataset at “Counting Triangles”, increasing file buffer size, for HDD.

[SSD2] just containers and io.file.buffer.size					
	Elapsed	Avg Map	Avg Shuffle	Avg Merge	Avg Reduce
io.file.buffer.size: 4KB	41mins, 8sec	25	359	339	538
io.file.buffer.size: 128KB	41mins, 9sec	24	361	331	554

Table 10.13: Performance difference for YouTube dataset at “Counting Triangles”, increasing file buffer size, for SSD2.

To have a generic idea of the impact of “customs” and the “containers” settings, we present in Tables 10.14 and 10.15, the relative performance of HDD and SSD2 for a large network, namely YouTube, which shows that HDD is a better beneficiary.

“Customs” difference to “Containers”				
	Avg Map	Avg Shuffle	Avg Merge	Avg Reduce
HDD	4.00%	-28.85%	-97.65%	-11.39%
SSD2	0.00%	-2.23%	-95.28%	-10.41%

Table 10.14: Percentage difference between “customs” and “containers” settings for YouTube dataset, at “Counting Triangles” algorithm.

“Customs” difference to “Containers”				
	Avg Map	Avg Shuffle	Avg Merge	Avg Reduce
HDD	-26.14%	-16.59%	-	-9.72%
SSD2	-18.83%	0.78%	-	4.36%

Table 10.15: Percentage difference between “customs” and “containers” settings for YouTube dataset, at “Mutual Friends” algorithm.

10.5.3.4 Results on calculating connected components

Comparing SSD1 to HDD and SSD2, the Connected Components algorithm (Table 10.16) seems to slightly favor the SSD1 for small datasets (first five ones), at reduce phase which is surprising and somewhat hard to explain, because SSD1 has theoretically inferior performance to SSD2.

	Avg Map			Avg Shuffle			Avg Merge			Avg Reduce		
	HDD	SSD1	SSD2	HDD	SSD1	SSD2	HDD	SSD1	SSD2	HDD	SSD1	SSD2
Brightkite	14	14	14	11	11	11	0	0	0	0	0	0
Amazon	104	106	103	34	34	34	0	0	0	74	61	62
Gowalla	27	26	26	10	10	10	0	0	0	14	14	16
DBLP	54	54	54	15	15	15	0	0	0	35	34	33
YouTube	126	124	123	14	14	14	0	0	0	101	96	98
YouTube 2	247	243	244	28	24	24	0	0	0	428	424	408
Flickr	170	168	167	30	19	20	0	0	0	309	314	304
LiveJourna l	353	380	322	104	143	45	1	0	0	665	682	651
LiveJournal 2	417	-	347	137	-	57	0	-	0	930	-	912
Orkut	456	-	324	552	-	154	295	-	231	1448	-	1204

Table 10.16: Sum of average times for each phase for the iterative jobs of "Connected Components".

However, we argue that the function of SSD1's TWC is quite successful. The generic pattern is that map, shuffle and reduce times are close for both disk types for these small datasets, contrary to what the current studies suggest.

When the size of data increases, e.g., for the datasets of Flickr and LiveJournal the magnetic disk takes the lead at reduce phase over SSD1, which is mostly characterized as "write" procedure for the Hadoop framework. SSD1 performs quite slowly at shuffle phase for the LiveJournal dataset, which again is attributed to the TWC delivering inferior performance. The SSD2 generally delivers great performance especially at map and shuffle phase, noticeably as the datasets' size increase. For the reduce phase, HDD falls behind SSD2, but not with a great margin.

To a have a better understanding of the reasons behind the above performance behavior between HDD and SSD2, we examined the details of CPU and disk utilization during the execution of the 1st iteration of the connected components algorithm on the largest of our networks, namely Orkut. Hadoop's default settings allowed the execution of up to 6 maps simultaneously. Thus the execution of Orkut dataset (input file of 14 blocks at HDFS) was executed in three waves of maps. The map phase is CPU intensive, hitting 100% utilization. High disk throughput is required as well, with the disk constituting system's bottleneck causing high CPU wait times, especially for HDD (Figure 10.7Left), where during map phase CPU utilization falls between map waves. Consequently using SSD2 provides better CPU utilization. Excessive disk usage appears at shuffle phase demonstrating each disk's capabilities (Figure 10.7Right–10.8). At reduce phase, SSD2 performs slightly better.

The experiments established that default Hadoop settings are not optimized for hard disks, and that the technology of SSDs might have dramatic impact upon their (expected) performance. Most significantly, we provided solid evidence that hard disks can be competitive to solid state disks for some I/O patterns, at least for the application field that we have investigated.

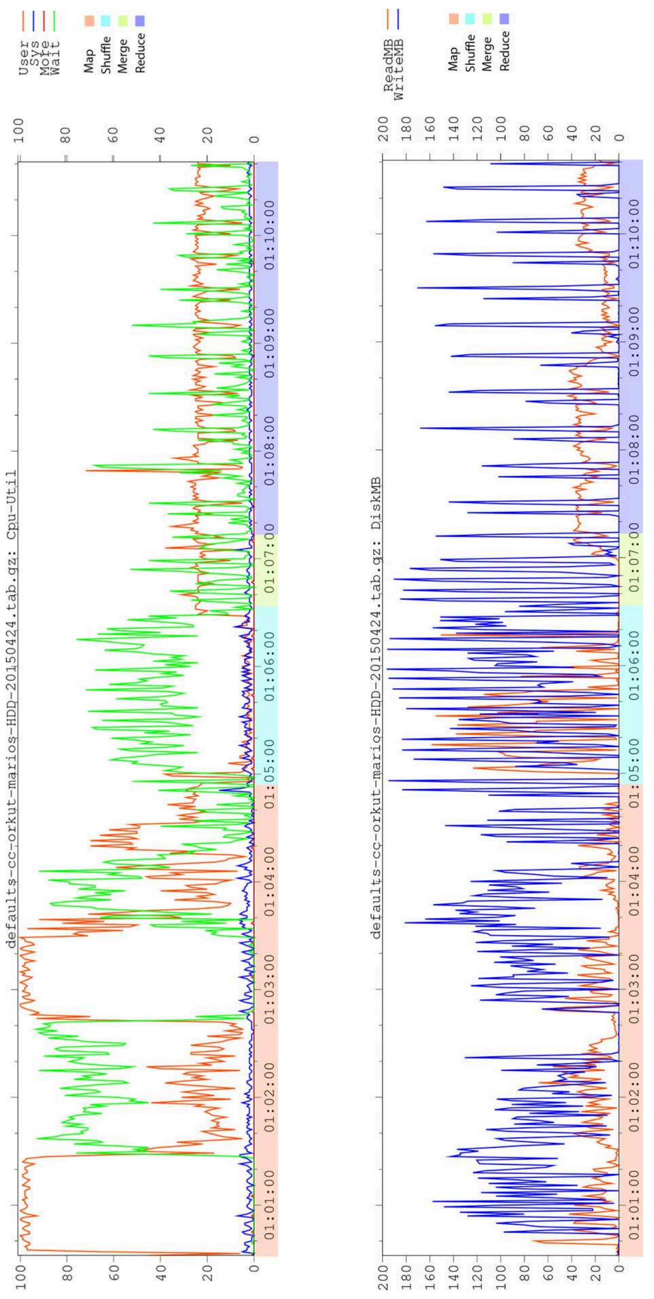


Figure 10.7: (Left) CPU utilization for Connected Components with Orkut, using HDD, 1st iteration isolated. (Right) Disk usage for Connected Components algorithm with Orkut, using HDD, 1st iteration isolated.

10.6 Conclusions

Hadoop platform is used for the processing of big data, especially to run analytics that is computationally intensive, such as social network analysis. Some tasks can be solved with a

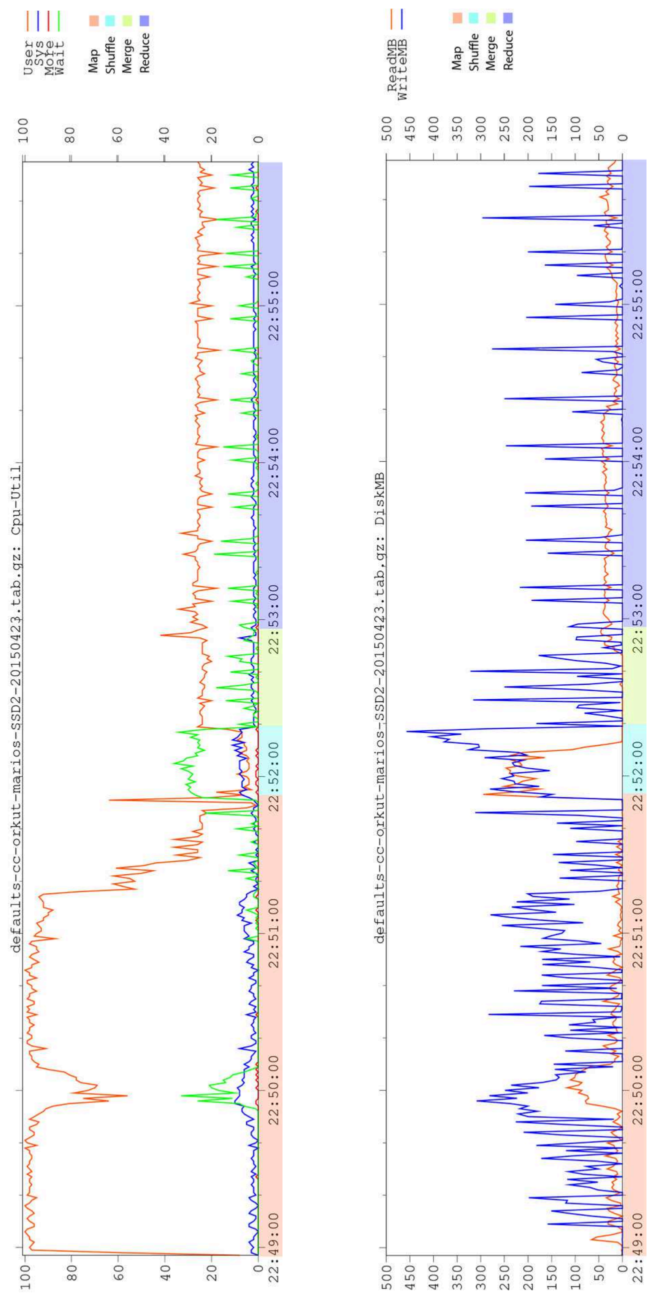


Figure 10.8: (Left) CPU utilization for Connected Components with Orkut, using SSD2, 1st iteration isolated. (Right) Disk usage for Connected Components algorithm with Orkut, using SSD2, 1st iteration isolated.

single or more consecutive and distinct jobs whereas others require iterative ones. Due to the SSD's provided substantial benefits over traditional hard disk drives, Hadoop administrators have started considering the addition or even replacement of the existing HDDs with SSDs. Yet, Hadoop's internal design - especially HDFS - doesn't appear to fully harness the potential of solid state drives.

In this empirical study, we compared the performance of solid state drives and hard disk drives for social network analysis. Three casual complex network analysis algorithms were used leaving space for the implementation and testing of many others, for even larger data sets.

A potential upgrade should be considered based on the tested applications' performance. In our tests SSDs didn't come out as the undisputed winner. There were noticed great performance fluctuations between the two SSDs. The second SSD performed significantly better. Otherwise, in many cases SSD1 and the magnetic disk came into a draw. Although SSD1 was slightly faster in many tests, *in some cases the magnetic disk outperformed the SSD1*. Even *compared to the faster SSD2, the magnetic disk provided competitive or faster times for reduce phase*, especially with the "mutual friends" algorithm.

Customizing Hadoop settings proves crucial. Magnetic disk's shuffle times can be significantly reduced. SSD's performance doesn't present further improvement. Nevertheless, HDD can't catch up with SSD's superior performance at shuffling. With tweaking merge-sort can be performed in less steps minimizing merge's phase times for both disk types, slightly favoring magnetic disk that would perform slower otherwise. For map phase both disk types can get similar performance improvement.

Overall, having no clear storage media winner, the paper suggests that the development of "application profilers" e.g., [20]–[22] that will try to predict the applications' read/write pattern (random/sequential) and then incorporation of them into the Hadoop architecture will help reap the performance benefits of any current or new storage media.

Chapter 11

Conclusions & Future Work

In this thesis we studied dynamical processes over complex networks, and focused on the spreading dynamics. We employed tools from graph theory and network science with aim to study network topology an uncover those node characteristics that play crucial role in spreading processes. We include a wide range of different network structures, i.e., single complex networks, probabilistic complex networks, multilayer complex networks and also vehicular ad hoc networks. Across all the development phases of our work, a common research approach has been followed. Particularly, all the proposed mechanisms where evaluated in widely adopted simulation environments in order to be consistent with the research community, be reproducible and thus provide solid proof of our findings.

Our work so far proved that the true spreading potential of network nodes cannot be "predicted" by merely measuring the number of connections (degree) incident upon the focal node. The reason behind this finding lies in the understanding that hub nodes positioned in the periphery of a network can not exert strong influence over a sufficiently large subset of the network nodes. On the other hand the k-shell decomposition of a network assigns a large number of nodes in the same shell, that is, the same (influence) importance. However our results illustrate that nodes positioned in the same shell quite often have significantly different spreading power. Although several new metrics based on k-shell where introduced in the literature addressing several shortcomings of the original algorithm, k-shell is based on global knowledge of the network topology and is thus unsuitable when faced with gigantic networks (millions of nodes and even more edges), incomplete knowledge of network topology and real time applications, etc. Our work introduced a hybrid method of node degree and k-core, that based solely on local knowledge of the network topology outperformed these state-of-the-art approaches by providing a more accurate ranking for the spreading potential of complex nodes.

Subsequently we focus on the multilayer network, i.e., networked systems where nodes may communicate through multiple type of connections. We generalize the well established h-index centrality in the domain of multiplex and interconnected networks by introducing a number of novel approaches that define the importance of a multilayer node. Likewise, the proposed methodology is based on local knowledge of network (and layer) connectivity, that is, at most two hop neighbor related information. We recognize and prove that a node able to exert strong influence over the multilayer network, must be well connected to as many layers as possible and interpret this attribute in the proposed schema for identifying influential spreaders in these complex systems. We employ a wide range of competing algorithms−and their respective generalizations to the multilayer domain−that are based on random walks, shortest paths, measures of connectivity, the k-shell etc., and prove via detailed simulations the superiority of the proposed technique in a wide range of real and generated multilayer networks.

Part of this dissertation focuses in the friendship paradox, that is *neighbor superiority* among the network nodes, and to its interpretation in the domain of complex systems. By casting our investigation in the context of the generalized friendship paradox we consider different topological node characteristics (by means of centrality), and empirically show that the paradox−at both network and node level−holds for an array of very popular centrality metrics not necessarily correlated to node degree. We further prove that the paradox intuition also applies to probabilistic measures such as the spreading power of a node, that is measured based on the most popular spreading models, i.e., the susceptible-infectious-recovered (SIR) and the susceptible-infectious-susceptible (SIS), for a range of diffusion probabilities near a network's epidemic threshold. The findings of this investigation can straightforwardly be used for designing better influential nodes detection algorithms, e.g., by refraining from selecting as initial spreading nodes those who are neighbors or for estimating the spreading capability of nodes using their friends' capability.

Focusing on the vehicular network we also studied diffusion processes over such immensely mobile and dynamic systems. We employ metrics from graph theory to accelerate the spreading process in the VANET. We highlight the importance of this unique network and its applications in the vehicular ecosystem, and furthermore investigate on the impact of infected (with marware/virus) vehicles in VANET protocols. We propose novel distributed methods for hindering the outspread of a virus based on network science methodologies, by triggering a negating spreading process to counter the effects of a malicious propagation. We separate the task of infection blocking from the task of disinfection; the latter is highly dependent on the kind of software that creates the infection whereas the former task can be performed in-situ in a distributed fashion with the cooperation of other vehicles and minimal use of fixed infrastructure. This study can be likened to node/link removal algorithms for blocking "contagions" in static complex networks. Our simulation results over a range of different scenarios and realistic parameters, indicate that the outspread of the virus can be significantly hindered, until an appropriate "cure" is distributed over cellular communications or physical treatment is administered. We further investigate on infected vehicles and the impact of their generated fake data in routing protocols. We employ various attack scenarios to deceive the system's decisions with aim to create traffic congestion in selected road segments. We urge for immediate attention in the infected VANET and its catastrophic results. We deploy a defense mechanism that is based in V2V, V2I and I2I communication to filter out spurious data running through the decision phases of our protocol. Our simulation results show that the proposed defense system successfully identified outliers and restored the protocol's performance to near normal behavior, i.e., as if no fake data were present.

Finally, we empirically study the potential benefits of utilizing SSDs (compared to HDDs) in the Hadoop ecosystem and answer the following question; ignoring any network biases and storage media cost considerations, do SSDs provide improved performance over HDDs for real workloads that are not dominated by either reads or writes? We employ our framework in social network analysis and particularly study three different problems directly related to spreading processes in social networks. The first algorithm deals with a very simple problem which is at the same time a fundamental operation in Facebook, Twitter, LinkedIn, etc., that of finding mutual friends. The second algorithm deals with a network-wide path-based analysis for finding connected components which finds applications in reachability queries, techniques for testing network robustness and resilience to attacks, epidemics, etc. The third algorithm is about counting triangles which is a fundamental operation for higher level tasks such as calculating the clustering coefficient, or executing community finding algorithms based on clique percolation concepts. Our work suggests that the development of "application profilers" that will try to predict the applications' read/write pattern into the Hadoop architecture will help reap the performance benefits of any current or new storage media

The advances in network technology predispose the increased complexity of our networked systems; nodes are able to communicate through multiple type of connections and thus facilitating communication through diverse networked environments, growth in colossal sized structures with millions of nodes and even more connections that require advanced handling and analysis, opportunistic connections of numerous wireless devices that play a fundamental role in dynamical processes, etc. All these considerations instruct that future networks hold a vast domain of yet undiscovered tasks and traditional network theory needs to appropriately adapt and evolve in order to embrace the newly and yet undiscovered needs of future networks. To this end the analysis of multilayer networks and dynamical processes on these structures will be a core part of our future directions. Understanding the peculiarities of each networked system and further combine those attributes in this multi-structure poses significant challenges. Among the different open problems to be solved, we highlight the following: (i) the need of setting up other metric concepts that could possibly affect relevant parameters of the systems and applications, such as the betweenness, community detection, etc., or broadly speaking the generalization of network science tools (such as centrality metrics) widely established in single networks to multilayer structures, with aim to to better understand the topology of these unique networks; (ii) gathering a better knowledge and understanding on the mathematical relationships that bind each respective layer separately–as a single network–and each layer component (e.g., each node or edge), and how those mathematical connections are measured and correlated on the whole multilayer structure; (iii) the study of diffusion processes to better understand and formally develop mathematical models that accurately project information propagation in these complicated systems and how diffusion processes on separate layers affect and develop multilayer diffusion; (iv) and finally the implementation of network generators based on observation of real multilayer structures (such as intra and inter layer degree distribution) to accelerate and foster new research challenges towards this domain. These consideration are only the start line of otherwise endless research directions towards understanding our ever evolving network structures and their applications in our everyday lives.

Appendix A

Materials and Methods

A.1 Spreading models

There is a lot of research interest in studying dynamic processes on large graphs, (a) blogs and propagations, (b) information cascades and (c) marketing and product penetration. These dynamic processes are all closely related to virus propagation, with many directly based on epidemiological models. A wide range of spreading models that simulate the spreading dynamics over complex networks is introduced in the literature. In this section we provide the details of the most widely spreading models adopted by the research community and employed throughout this dissertation. For more details please refer to [106].

A.1.1 Susceptible-Infectious-Recovered (SIR)

From the perspective of SIR, the population (the network nodes) is subdivided into three groups, the susceptible (ignorant) group where nodes are ignorant of the emergence for example of a virus, meme, rumor, etc., and are potential adopters; the infectious group composed from a set of nodes that are initially incentivized (infected) to spread a product (virus); and finally the recovered group (stiflers) that consists of nodes that are no longer interested in the corresponding propagation (or vaccinated against the particular virus). In the epidemic spreading, each time an infected node contacts with a susceptible node, there is a chance that the susceptible node gets infected. Based on this fact, independent interaction models assume that each interaction results in contagion with independent probability. Particularly, whenever a susceptible person j is exposed to an infected node i, j becomes infected with probability p_{ij}. We assume that the spreading probability for all node pairs is the same, i.e., $p_{ij} \equiv \beta$. On the other hand an infected node enters the recovered state with probability μ (Figure 1.3). Specifically, the distinct node states are:

- the susceptible (**S**) state, where nodes can be infected (influenced) if they are connected to an infected node.

- the infectious (**I**) state, where nodes try to infect (influence) their susceptible neighbors and succeed with probability β.

- the recovered (**R**) state, where nodes cannot be infected (influenced).

SIR immunizes nodes (R state) and thus measures the penetration for example of a virus (or rumor, product, meme, etc.) in a networked environment. It is widely employed in this thesis to quantify the spreading power of a node, i.e., its influence over the remaining network nodes. Initially a specific node (or a set of nodes), the node of interest (focal node) is infected, while the remaining nodes are in state S. SIR unfolds in discrete steps. In each step all infected nodes try to infect their susceptible neighbors and succeed with probability β. Immediately after, without loss of generality, the infected node with enter the R state, i.e., $\mu = 1$. The spreading steps unfold in subsequent rounds until there is no node in state I. Thus, *the influence exerted by the initially infected node* is quantified by the number of nodes in the R state at the end of SIR.

A.1.2 Susceptible-Infectious-Susceptible (SIS)

SIS is very similar to SIR. SIS offers no immunization for the network nodes, i.e., the recovered state is excluded. SIS assumes that agents (the nodes) can only exist in the two remaining discrete states: susceptible or healthy (**S**) and infected (**I**). At each time step an agent tries to infect its susceptible neighbors and (likewise in SIR) succeeds with probability β, whereas infected nodes are “cured” by rate γ, i.e., return to the susceptible state (Figure 1.3). Therefore, agents can run stochastically through the cycle, susceptible−infected−susceptible. SIS measures the capability of a virus/meme/product to preserve itself with the network, i.e., become epidemic and constantly keep infected (interested) a large subset of network nodes, or fail, and die out quickly. In a similar fashion, in the initial step a node (or a set of nodes) is set in state I. In the sequence the transitions between the node states are unfolding with respect to the predefined probabilities. SIS terminates when a relatively fixed number of nodes remains in state I as the spreading steps unfold (also known as equilibrium phase) or the diffusion dies out and all nodes reside in state S. Thus, *the influence exerted by the initially infected node* is quantified by the number of nodes in the I state at the end of SIS.

A.2 Centrality Metrics

A wide range of centrality metrics have been employed throughout this dissertation and served as competitors and as sources of inspiration for our work. These metrics range from geodesics and random walks to measures that quantify the coreness and connectivity of the network nodes and so on. It has been shown in the literature that the topological characteristics of a node play a crucial role in the spreading dynamics and the influence exerted over the network. Hence the research community is focused in many such methods that encompass different attributes of the agent-nodes. For coherency we briefly present only but a fraction of these centrality metrics that are widely used in related processes and the core of this dissertation. We present each metric for unweighted networks, however their implementation to weighted structures is straightforward.

[Degree] Degree is a simple centrality measure that counts how many neighbors a node has, i.e., immediate (one hop) connections. If the network is directed, we have two versions of the measure: in-degree is the number of in-coming links, or the number of predecessor nodes; out-degree is the number of out-going links, or the number of successor nodes.

[PageRank] PageRank is an algorithm used by Google Search to rank websites in their search engine results. PageRank was named after Larry Page, one of the founders of Google. PageRank is a way of measuring the importance of website pages; in sort an important page (node) is one that is point by many other and important nodes. For a node x:

$$PageRank(x) = \frac{1-\delta}{N} + \delta \sum_j a_{jx} \frac{PageRank(j)}{L(j)} \tag{A.1}$$

where a_{jx} is 1 if j links to x and 0 otherwise, $L(j) = \sum_x a_{jx}$ s the number of neighbors of node j (or number of outbound links in a directed graph) and δ is the damping factor.

[Betweenness] Betweenness centrality quantifies the number of times a node acts as a bridge along the shortest path between two other nodes. It was introduced as a measure for quantifying the control of a human on the communication between other humans in a social network. For a node x:

$$betweenness(x) = \sum_{s \neq x \neq t \in V} \frac{\sigma_{st}(x)}{\sigma_{st}} \tag{A.2}$$

where σ_{st} is total number of shortest paths from node s to node t and $\sigma_{st}(x)$ is the number of those paths that pass through x.

[Closeness] In a connected graph, the normalized closeness centrality (or closeness) of a node is the average length of the shortest path between the node and all other nodes in the graph. Thus the more central a node is, the closer it is to all other nodes. Closeness was defined by Bavelas (1950) as the reciprocal of the farness. For a node x:

$$closeness(x) = \frac{1}{\sum_y d(y,x)} \tag{A.3}$$

where $d(y,x)$ is the shortest distance between nodes x and y.

[k-core] K-shell (or k-core) decomposition of a network graph(Figure A.1) is performed iteratively. The first step involves removing all degree-1 nodes, along with their link, and indexing these as k = 1, i.e., nodes in the first shell (1-shell) also known as peripheral nodes. In the resulting graph (the subgraph of the original network after the first step of removal), all nodes with remaining degree 1 are also considered to have k = 1 and are again pruned. The process is repeated until there are no remaining nodes of degree 1. In the sequence and in a similar fashion, degree-2 nodes are removed and indexed as 2-shell nodes. Generally all nodes with i or fewer connections are iteratively removed; these nodes are indexed i-shell nodes.

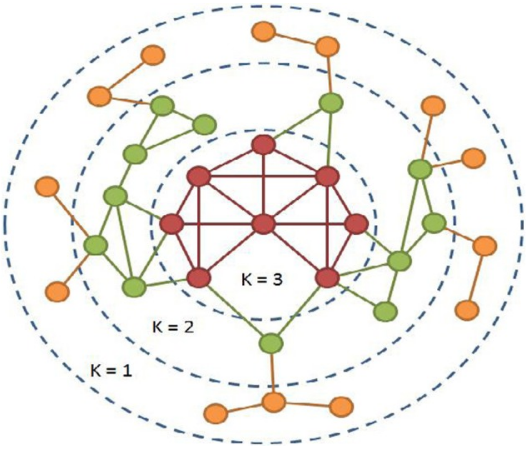


Figure A.1: Example of the k-shell decomposition method.

A.3 Performance Evaluation

Kendall's τ

We use the Kendall's tau rank correlation coefficient τ to demonstrate how a specific ranking is correlated to the ranking induced by the spreading ability the nodes. The Kendall's tau coefficient considers the list produced by sorting the nodes according to their spreading power, and the list produced by sorting the nodes according to the value of function Y, where Y_i are the node's i score assigned by a specific centrality measure, e.g., PageRank, PCI, etc. Note that both lists are of the same size i.e., n. Then, the τ value can be computed as follows:

$$\tau = \frac{n_c - n_d}{n(n-1)/2} \tag{A.4}$$

where n_c is the number of concordant pairs, and n_d is the number of discordant pairs. The denominator is the total number of pairs of n items in the lists. For each pair of items in the list, we determine if the relative rankings between the two lists match. For pair of nodes (i,j), if node

i is ranked above (or below) node j in both lists SP and Y, then the pair is called *concordant*. Otherwise, it is called *discordant*. Clearly, $-1 \leq \tau \leq 1$. If $\tau = 1$, then the two rankings are in perfect agreement; if $\tau = -1$, then one ranking is the complete reverse of the other.

APPENDIX B

APPENDIX OF CHAPTER 4

B.1 Multilayer network generator

Multilayer network generator

Our synthetic multilayer network generator can define:

- How many interlinks, i.e., inter-neighbors, a node may have.
- How those links are distributed over the layers.
- How links are distributed in each specific layer.

In other words we are able to synthesize the distribution in the number of the interconnections per node, i.e., the inter-degree of nodes, how those links are distributed to the different layers, and finally their distribution in a specific layer. Controlling all such parameters for the interconnections, allow for the creation of a diverse multilayer environment of interconnected entities with varying characteristics. For instance, it may be of interest to have a uniform distribution for the inter-degree of the nodes, or, to apply some power law distribution in order to have several *layer-hub-nodes*, i.e., a few nodes with many links to the different layers, while the remaining nodes have limited interconnectivity. Similarly it may be of interest to have a uniform distribution of those links to the different layers, or, to have layers that accumulate the majority of those interconnections, i.e., *hub-layers*. Finally we can apply the same policy to nodes within a layer, i.e., a few nodes gather most of the interconnections, while the rest have narrow interconnectivity.

We apply the Zipfian distribution in our interconnectivity generator. The Zipfian's law depicts the frequency of occurrence over a range of values, e.g., the frequency (or rarity) of high inter-degree nodes. The desired skewness (or uniformity) is managed by the parameter $s \in (0,1)$.

Increasing in s implies increase in skewness, which in our example is interpreted as rarer high inter-degrees, whereas values closer to zero imply closing to uniform distribution. Hence, in our framework we apply three distinct Zipfian laws, one per parameter of interest:

- $s_{degree} \in (0,1)$ in order to generate the frequency of appearance of highly interconnected nodes.

- $s_{layer} \in (0,1)$ in order to choose how frequently a specific layer is selected.

- $s_{node} \in (0,1)$ in order to choose how frequently a specific node is selected in a specific layer.

Then, we can decide the range of values for the different distributions. For s_{layer} and s_{node} the selection is straightforward since all layers and all nodes within a layer must be available options. Note that the different layers are allowed to have different preferences, i.e., skewness towards different network-layers. For example nodes in layer A may be skewed towards layer B, nodes in layer C may prefer nodes within layer D, etc. Following the review of [61] we understand that inter-connections are rarer than the intra-connections. In our simulations we limit the inter-degree of nodes within (0, $d \cdot log_2 \sum_i V_i$) for all i = 1,2,..N layers where d = 1, 2, 3 or 4. In our experiments, we applied the notation SLN$_d$(s_{degree}, s_{layer}, s_{node}) in order to refer to the different generated networks. Figures B.1 and B.2 exemplify the distribution for the out inter-degree of SLN$_2$(0.3, 0.3, 0.3), SLN$_2$(0.8, 0.3, 0.3), DLN$_2$(0.3, 0.3, 0.3) and DLN$_2$(0.8, 0.3, 0.3) with $d = 2$. We observe that when $s_{degree} = 0.8$ the majority of nodes is at the lower values of inter-degree, whereas when $s_{degree} = 0.3$, as expected, we obtain a broader distribution.

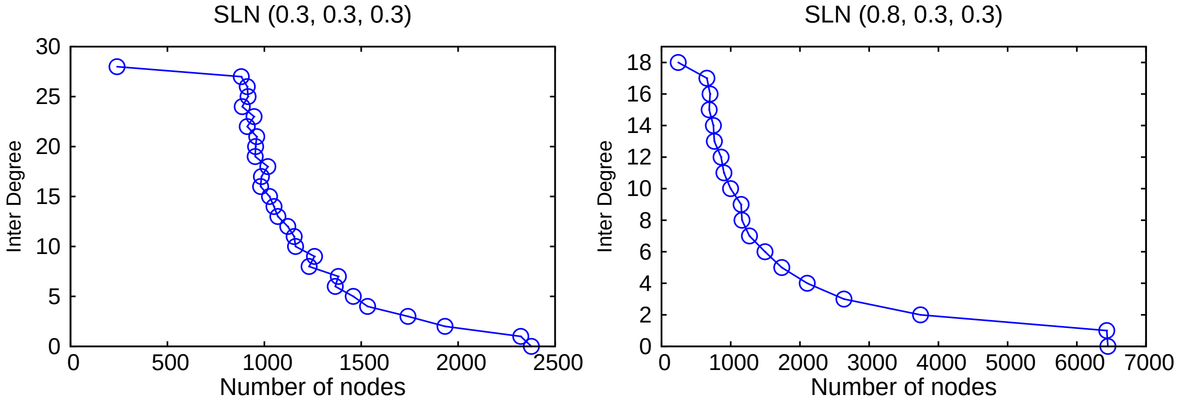


Figure B.1: Out inter-degree distribution for the SLN networks when $d = 2$.

Network properties

Figure B.3 illustrates the in-out degree of the real multiplex networks. Figure B.4 illustrates the out-degree (k_{out}) distribution of the different multilayer networks mentioned in section 4.4.4.2.

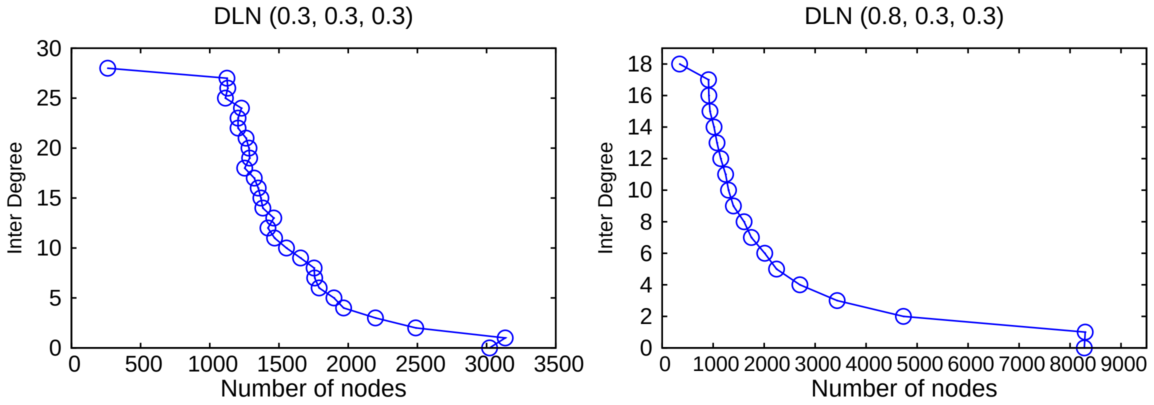


Figure B.2: Out inter-degree distribution for the DLN networks when $d = 2$.

All the Gnutella networks have similar k_{out} distribution, and thus we show here only p2p-Gnutella04's out-degree distribution.

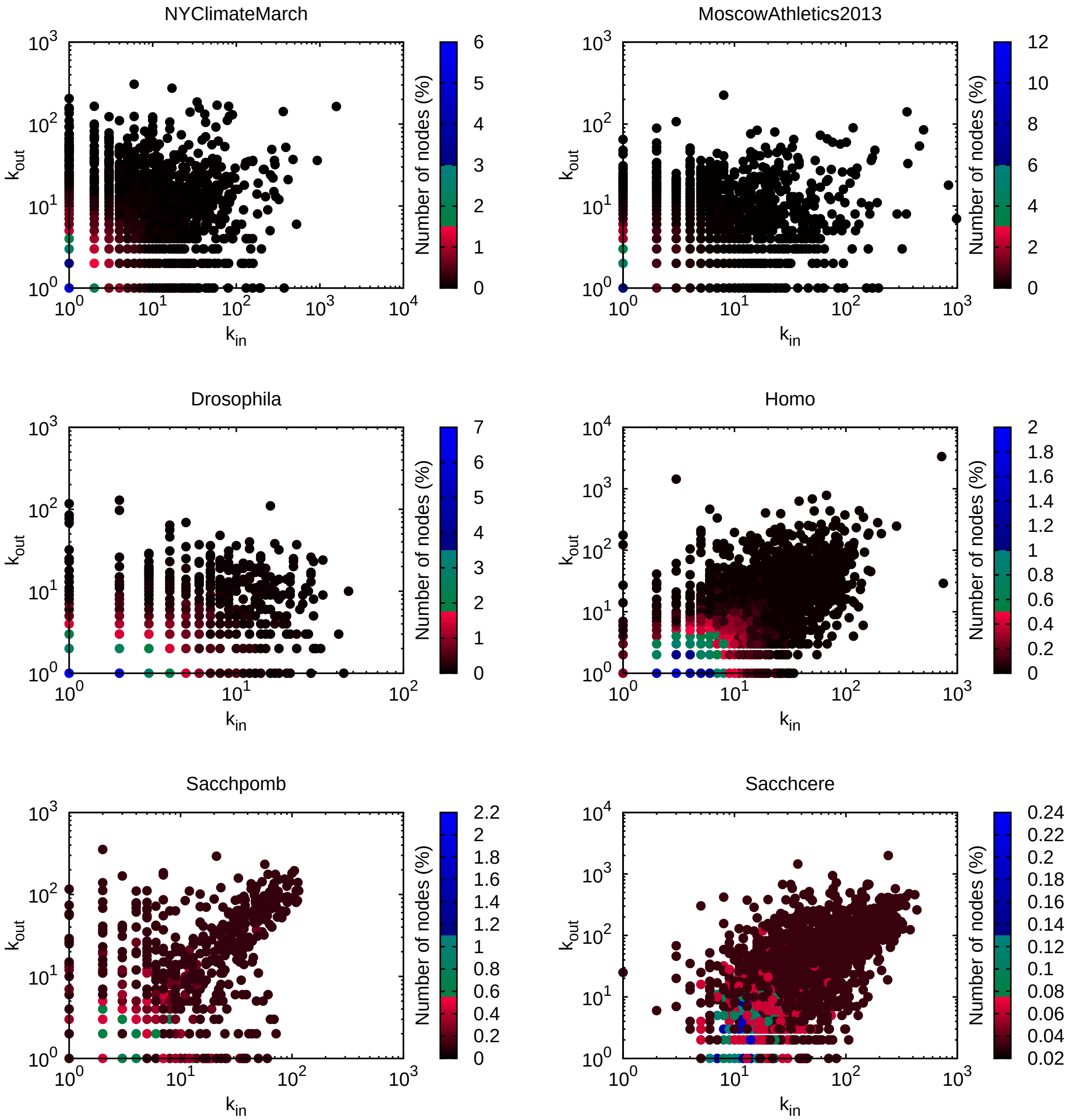
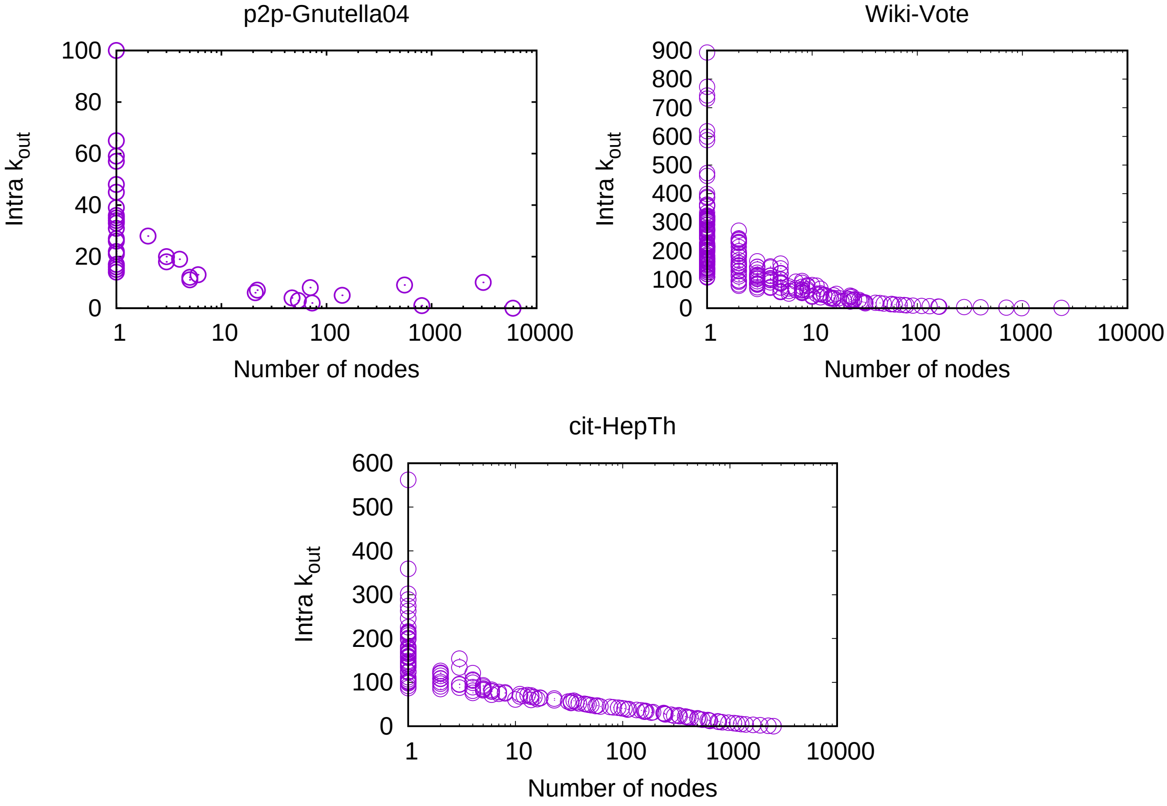


Figure B.3: Distribution of in-out degree for the evaluated networks. Colored dots illustrate the percent of network nodes with the specific pair of (k_{in}, k_{out}) values.



p2p-Gnutella04

Wiki-Vote

cit-HepTh

Figure B.4: k_{out} distribution of the layers for the semi-synthetic networks.

Appendix C

Supplementary for “On neighboring nodes’ relative power of influence"

C.0.1 Detailed experiments on the centrality paradox at the network level

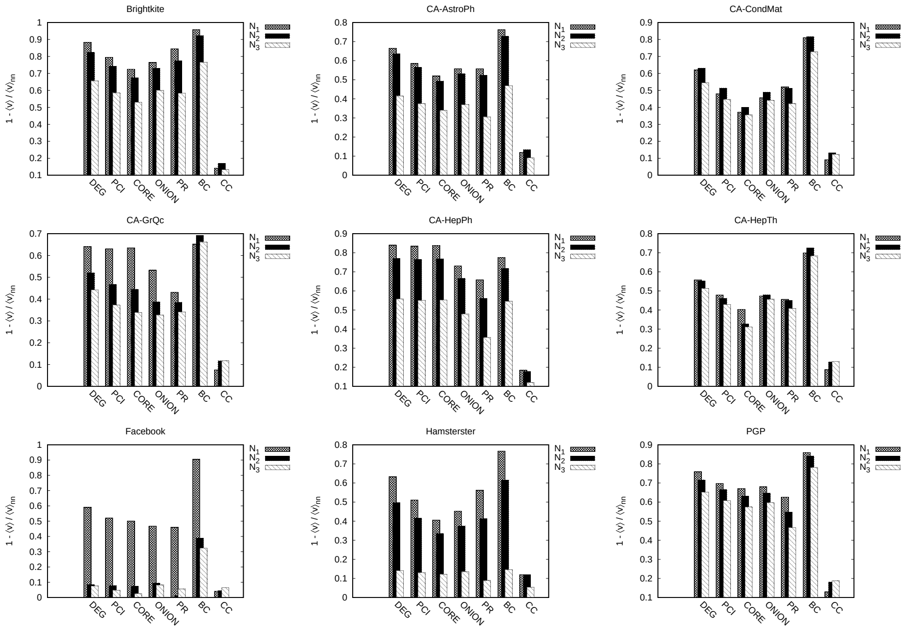


Figure C.1: Paradox evaluation at the network level for all centralities, all neighborhoods and all networks. The y-axis illustrates the ratio $\langle c_{neigh} \rangle / \langle c \rangle$ normalized to all neighborhoods (N^1, N^2 and N^3). Negative values indicate that the network level paradox does not hold. It can be observed that moving from N^1 to N^2 favors the paradox, i.e., $\langle c_{neigh} \rangle / \langle c \rangle$ increases (i.e., strengthens the paradox) in most of the illustrated networks. Extending the evaluated neighborhood one more hop (to N^3) illustrates a decreasing trend (weakens).

C.0.2 Detailed experiments on the centrality paradox at the individual level

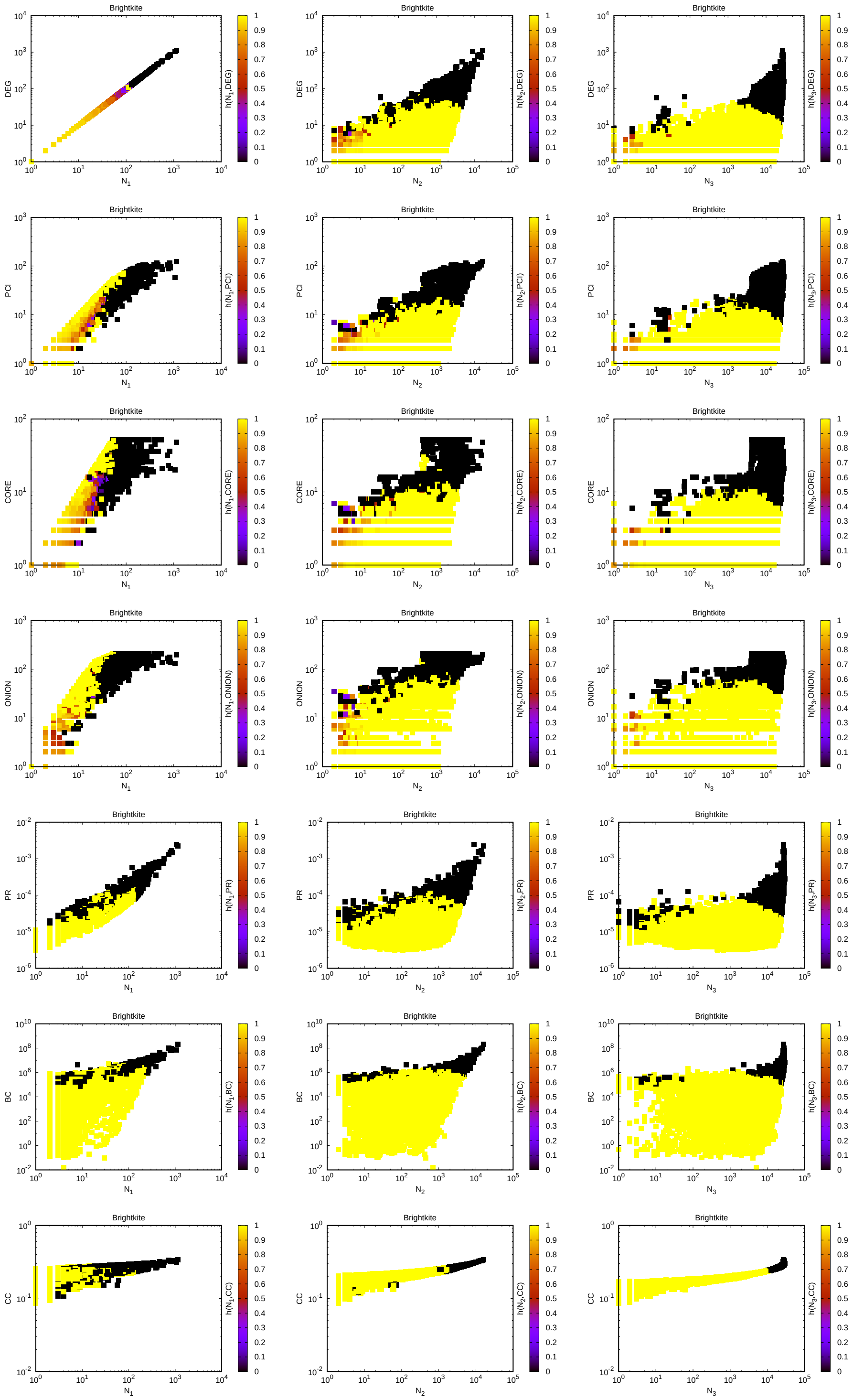


Figure C.2: Individual level centrality paradox for the *Brightkite* network in N_1, N_2 and N_3.

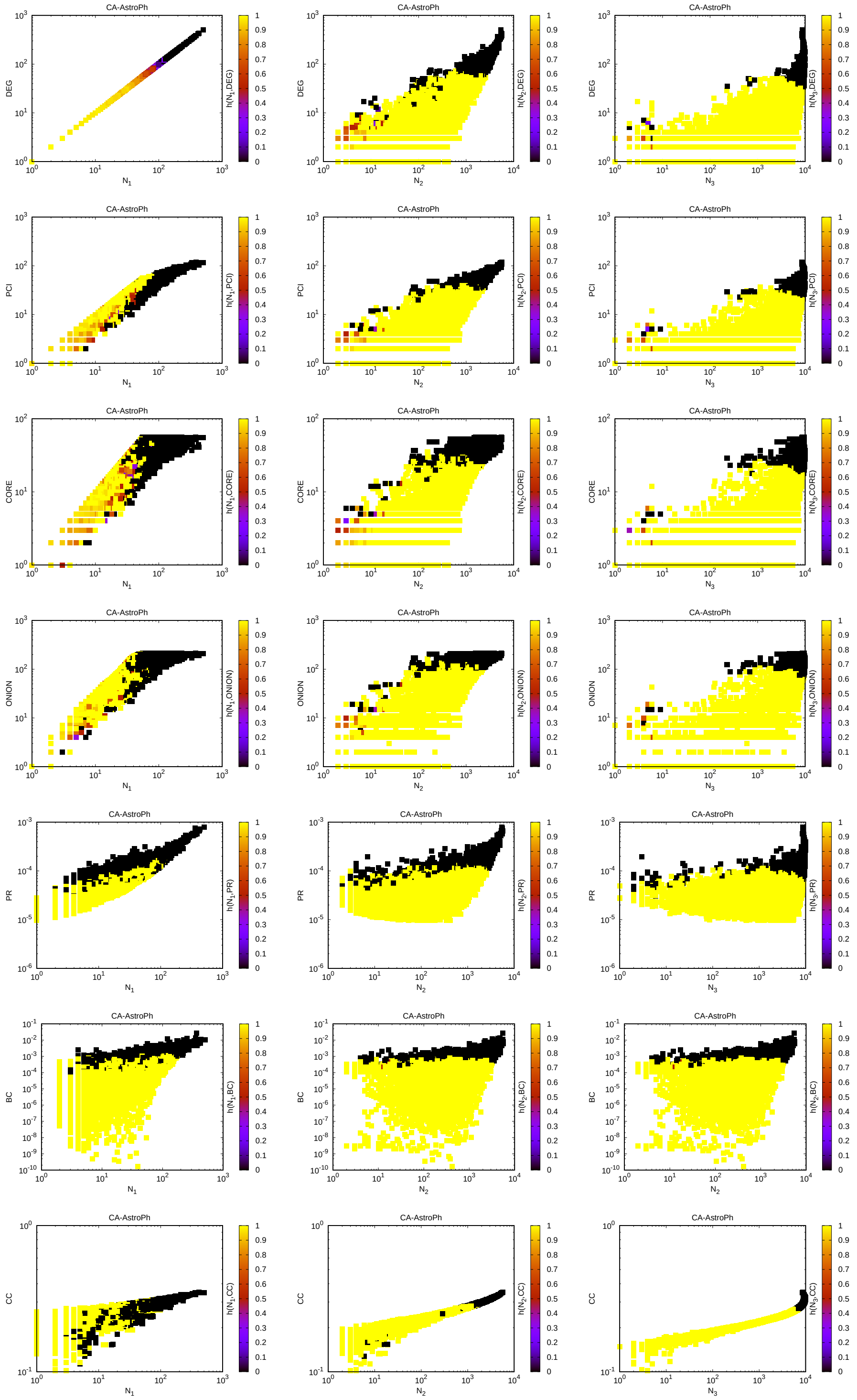


Figure C.3: Individual level centrality paradox for the *CA-Astroph* network in N_1, N_2 and N_3.

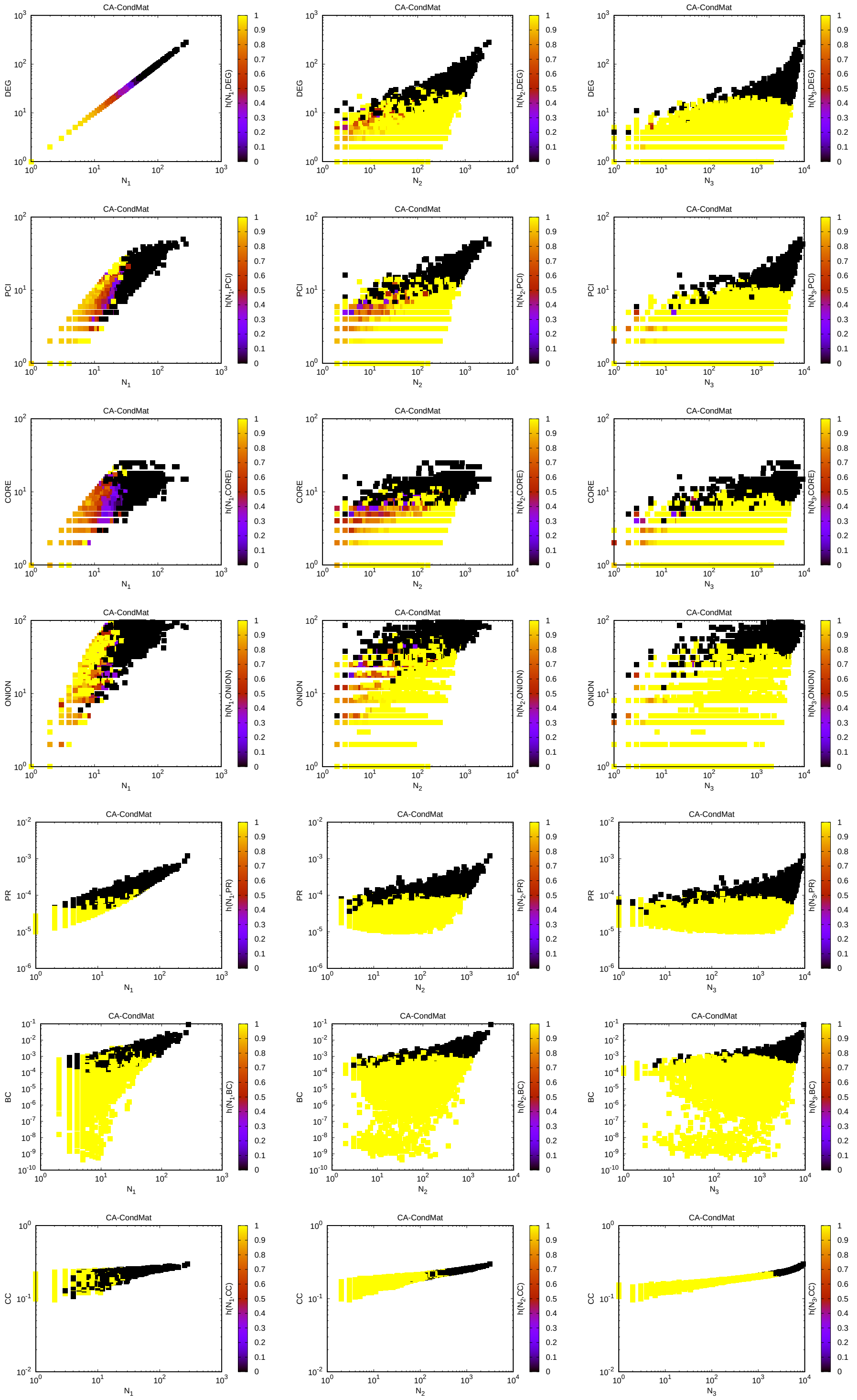


Figure C.4: Individual level centrality paradox for the *CA-CondMat* network in N_1, N_2 and N_3 neighborhoods.

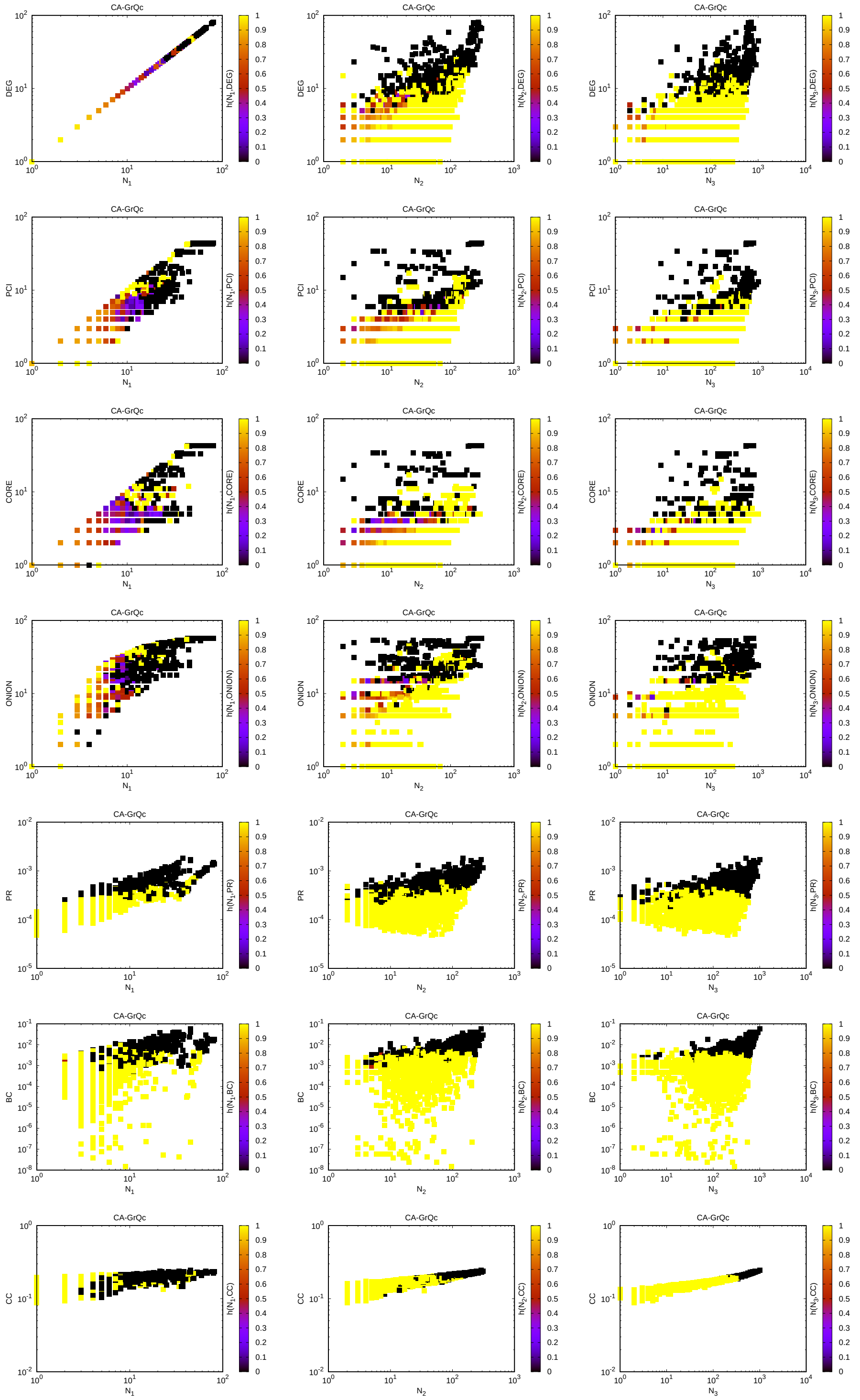


Figure C.5: Individual level centrality paradox for the *CA-GrQc* network in N_1, N_2 and N_3 neighborhoods.

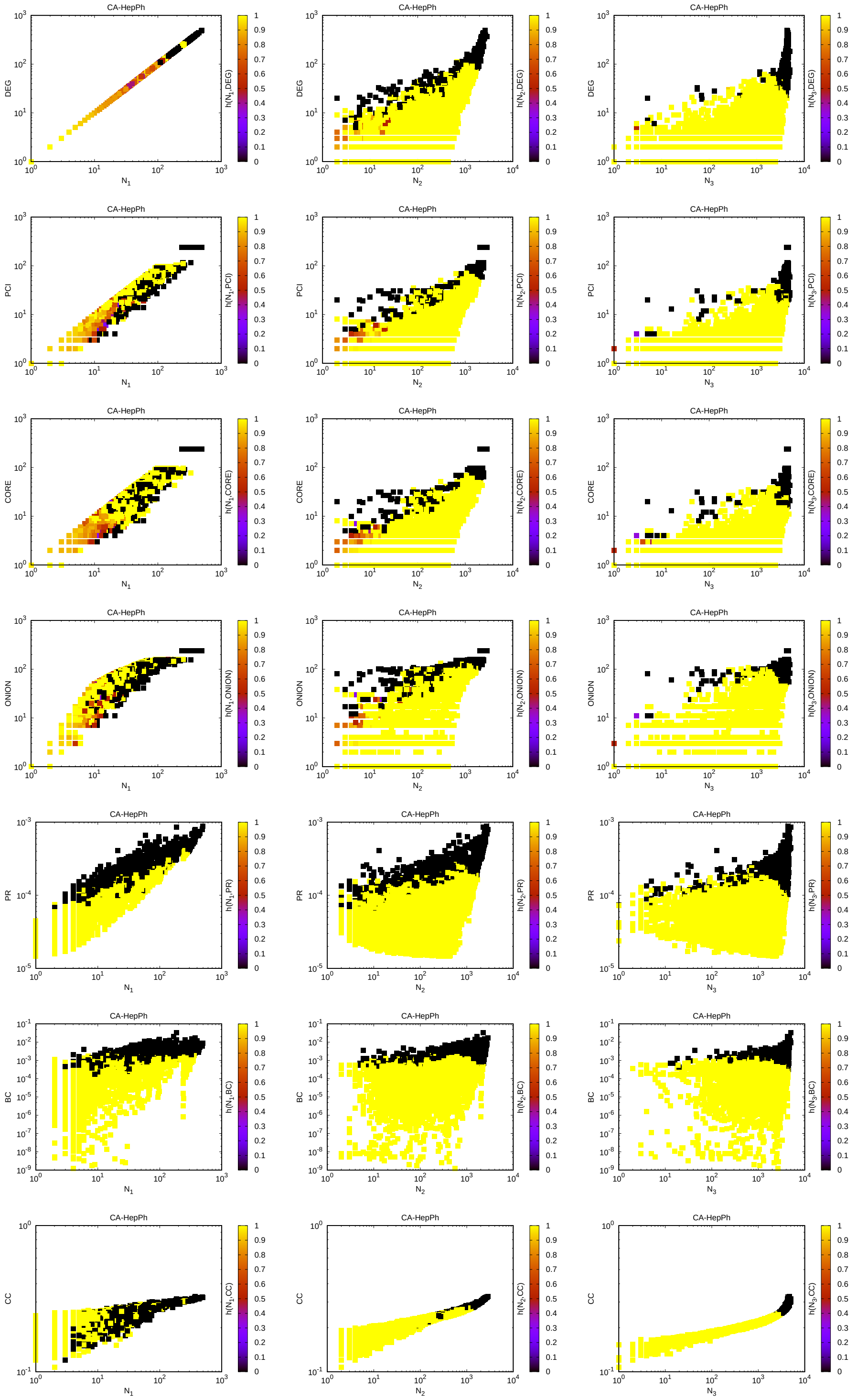


Figure C.6: Individual level centrality paradox for the *CA-HepPh* network in N_1, N_2 and N_3 neighborhoods.

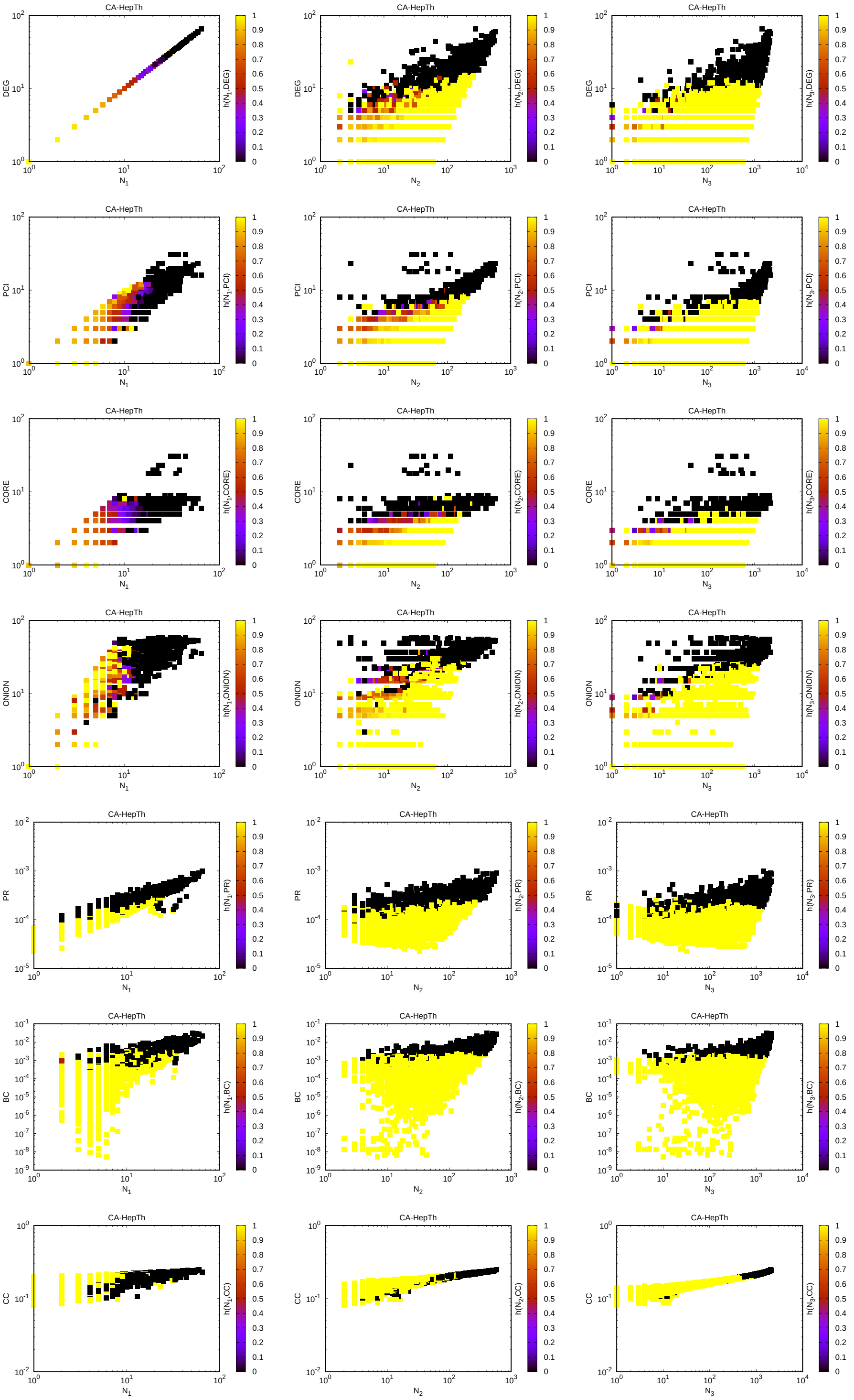


Figure C.7: Individual level centrality paradox for the *CA-HepTh* network in N_1, N_2 and N_3 neighborhoods.

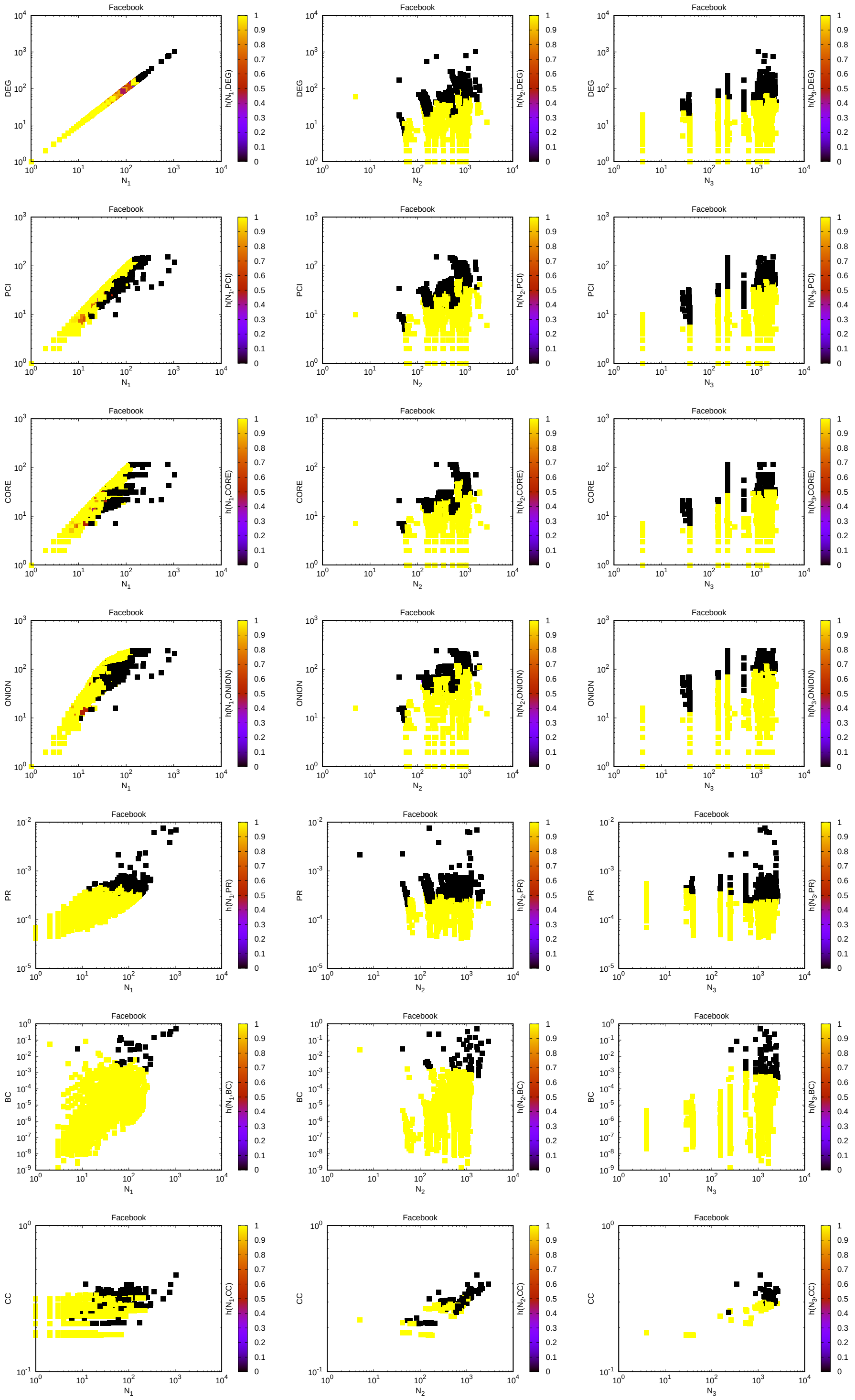


Figure C.8: Individual level centrality paradox for the *Facebook* network in N_1, N_2 and N_3 neighborhoods.

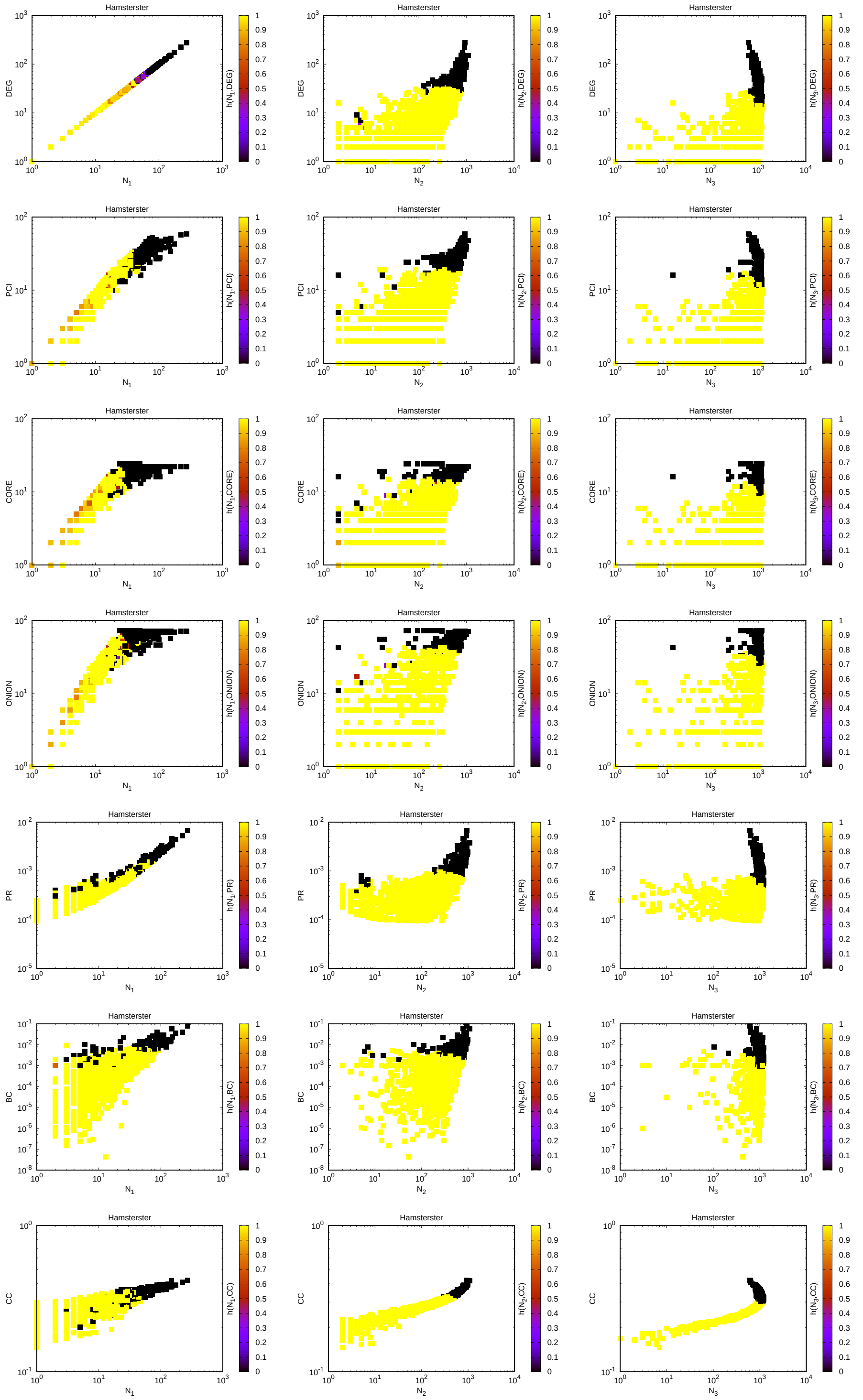


Figure C.9: Individual level centrality paradox for the *Hamsterster* network in N_1, N_2 and N_3 neighborhoods.

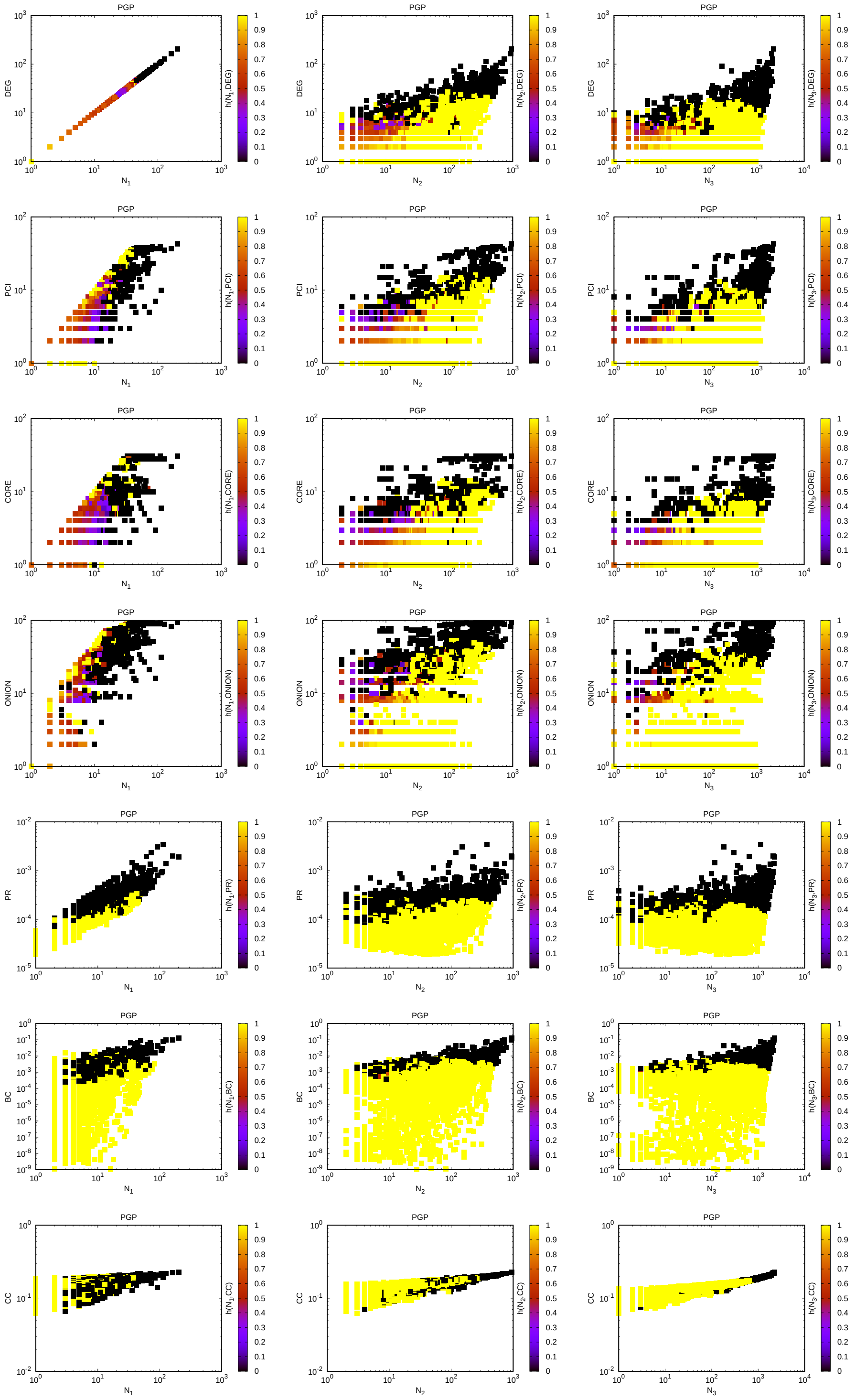


Figure C.10: Individual level centrality paradox for the *PGP* network in N_1, N_2 and N_3 neighborhoods.

C.0.3 Detailed experiments for the blocking application under the SIR model

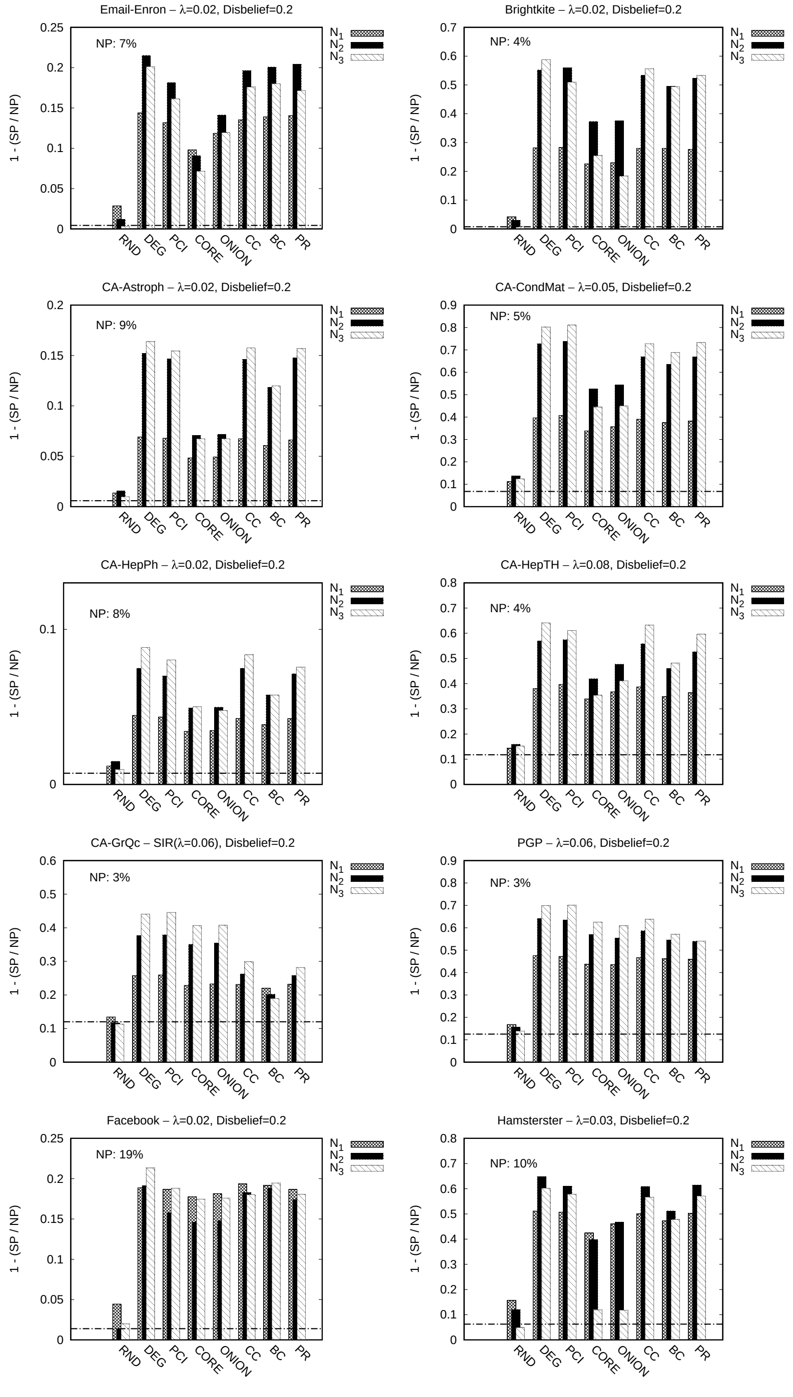


Figure C.11: Blocking the outspread of misinformation under the SIR spreading model for all networks. NP denotes the fraction of influenced nodes when there are no active blockers.

C.0.4 Detailed experiments regarding the spreading application for the SIR model

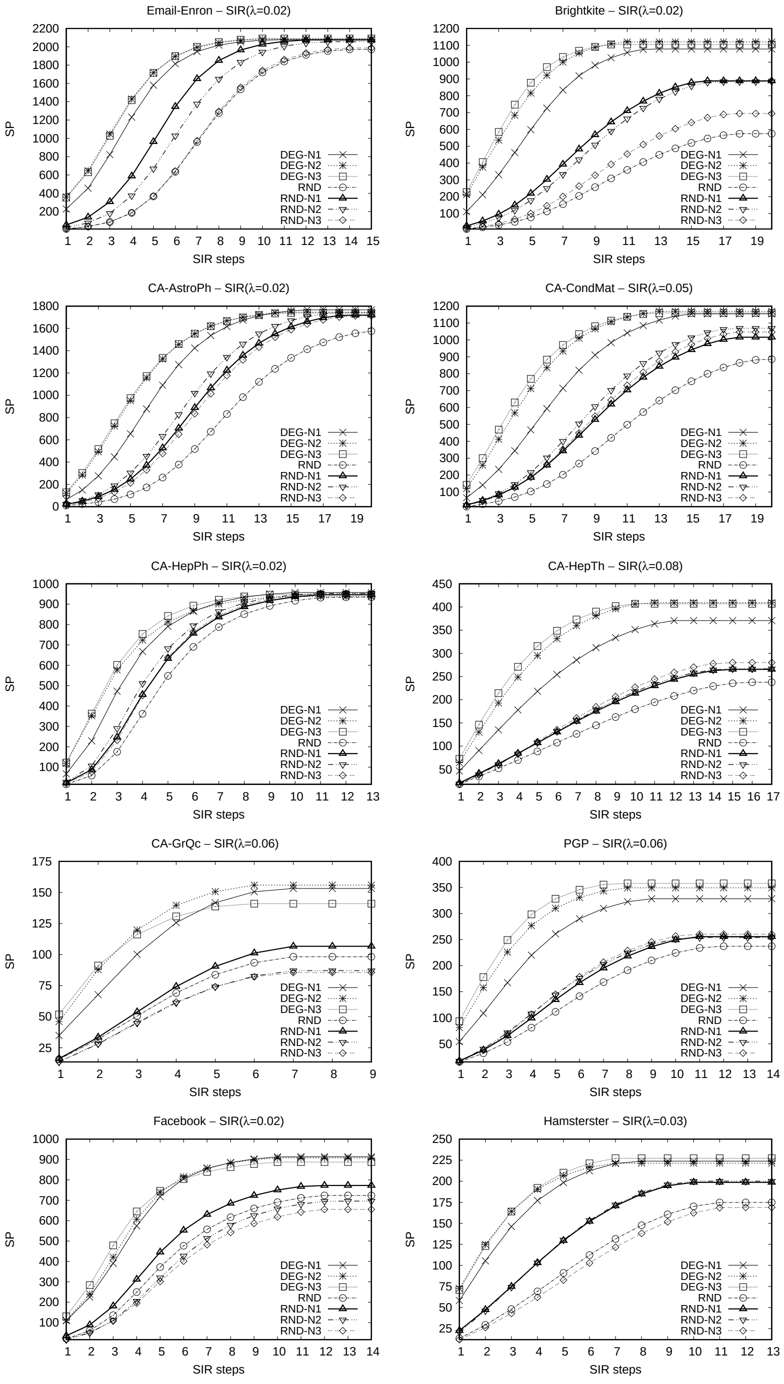


Figure C.12: Influence maximization for all networks under the SIR spreading model with cascade initiators biased towards the highest DEG nodes from N_1, N_2 and N_3 of RND.

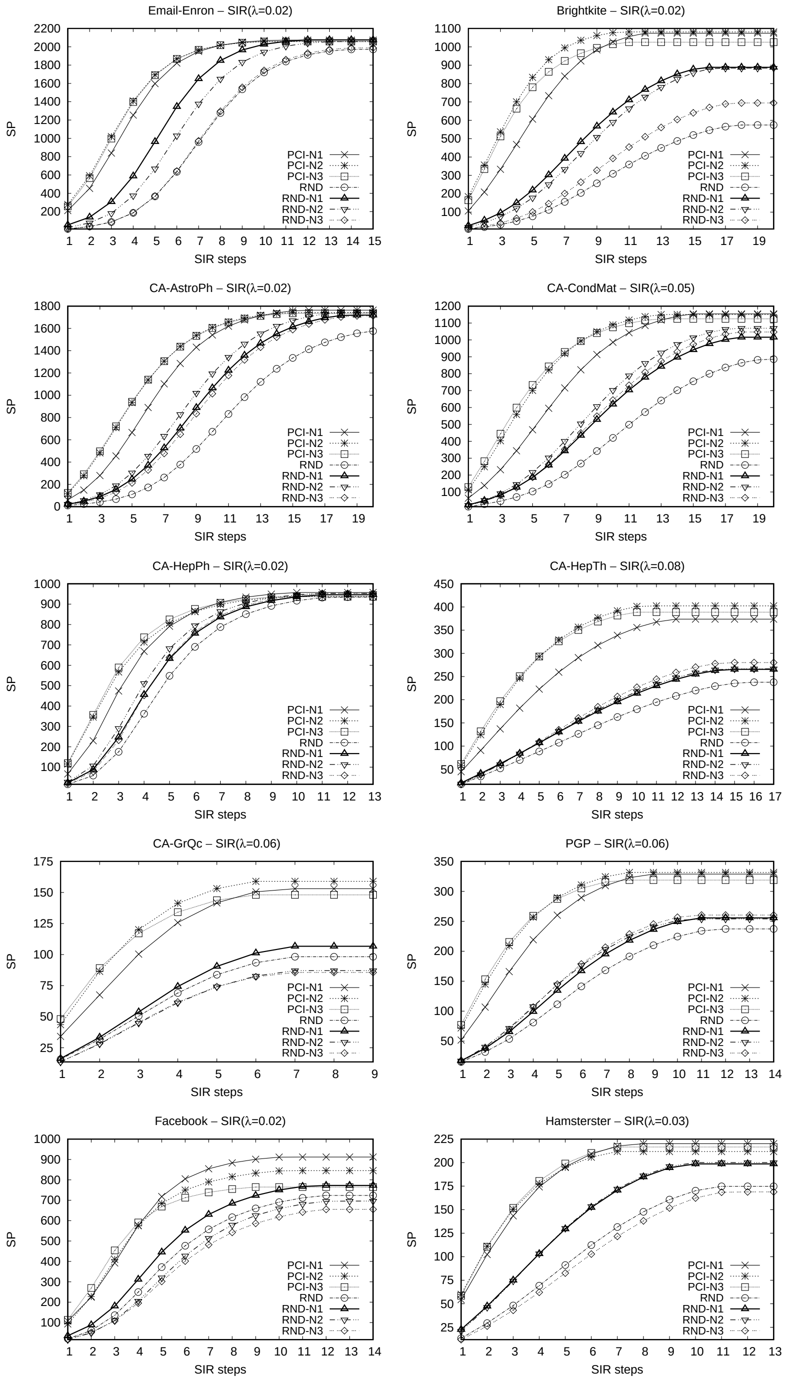


Figure C.13: Influence maximization for all networks under the SIR spreading model with cascade initiators biased towards the highest PCI nodes from N_1, N_2 and N_3 of RND.

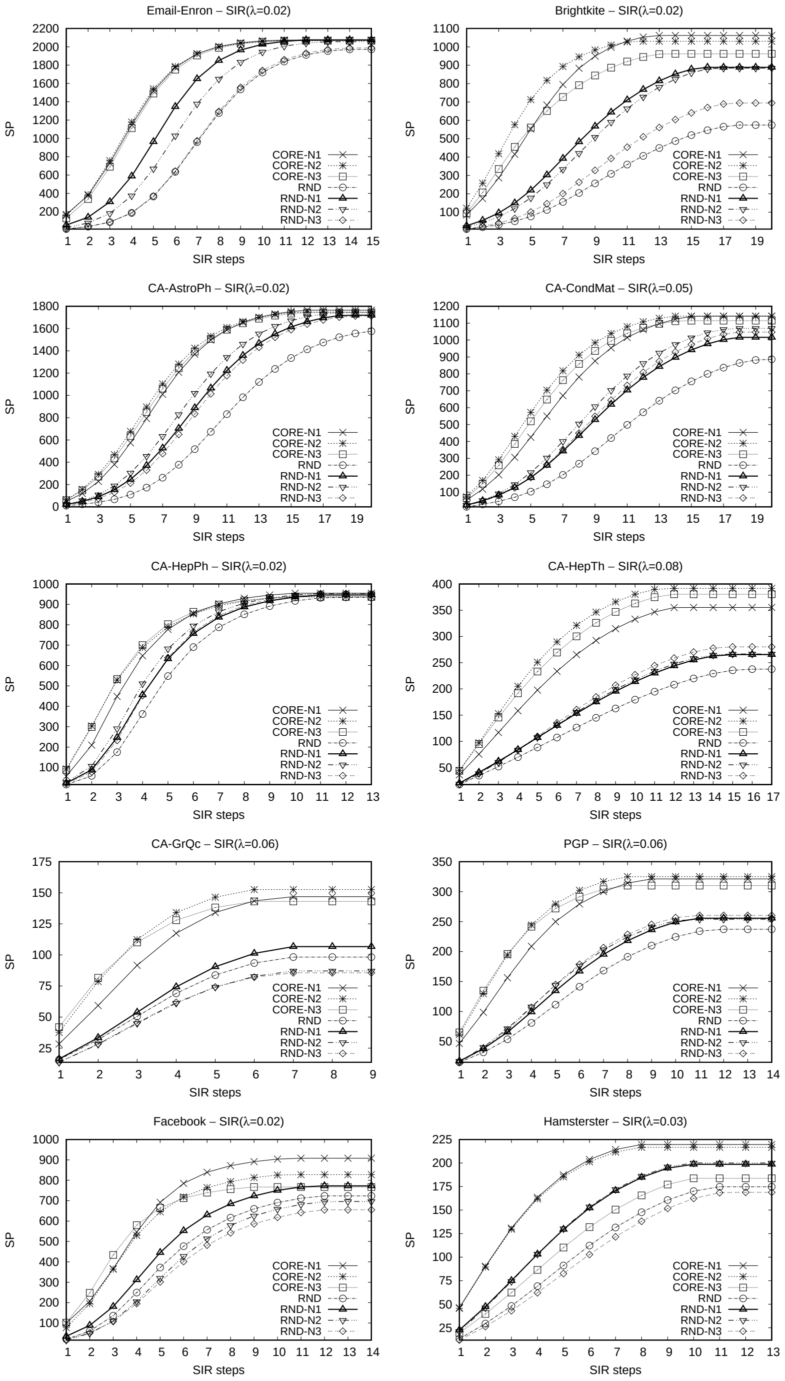


Figure C.14: Influence maximization for all networks under the SIR spreading model with cascade initiators biased towards the highest CORE nodes from N_1, N_2 and N_3 of RND.

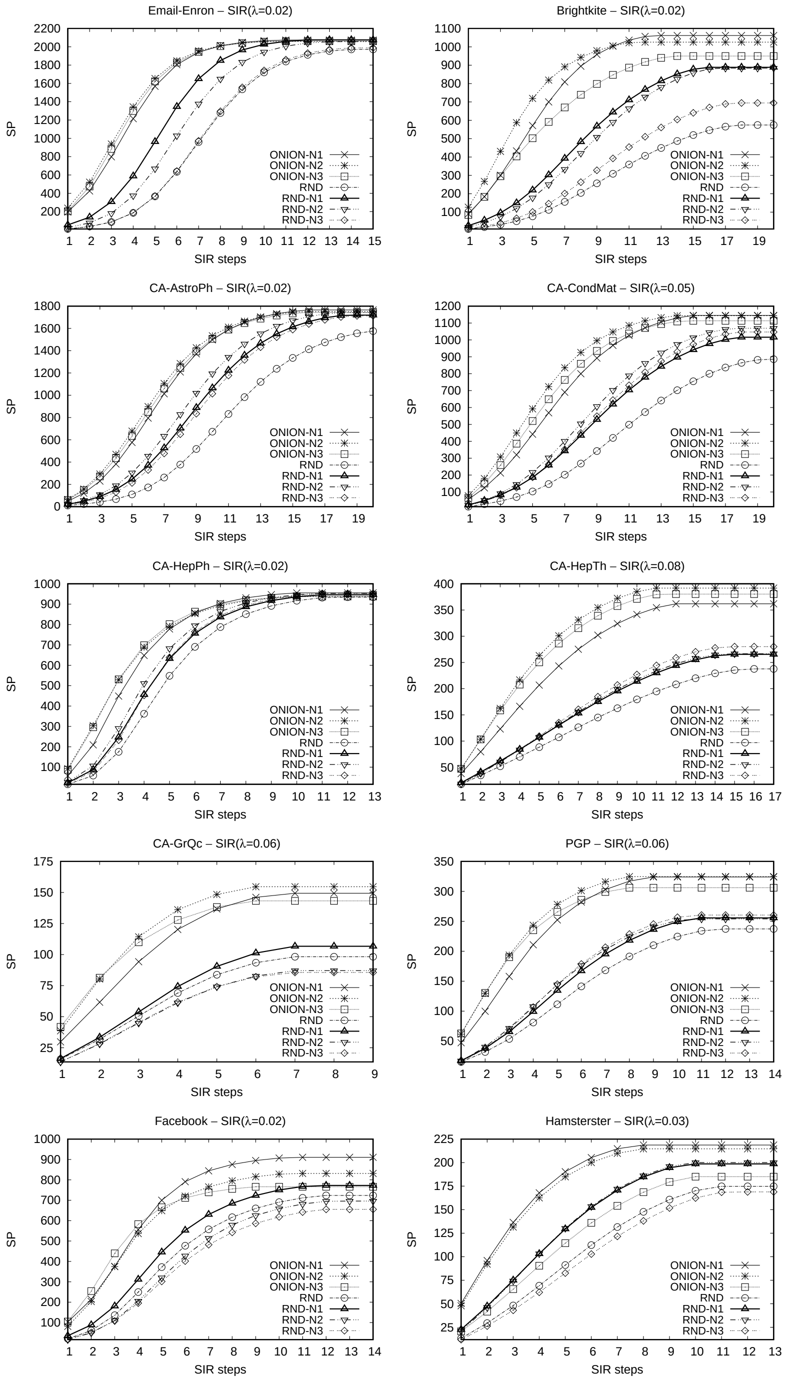


Figure C.15: Influence maximization for all networks under the SIR spreading model with cascade initiators biased towards the highest ONION nodes from N_1, N_2 and N_3 of RND.

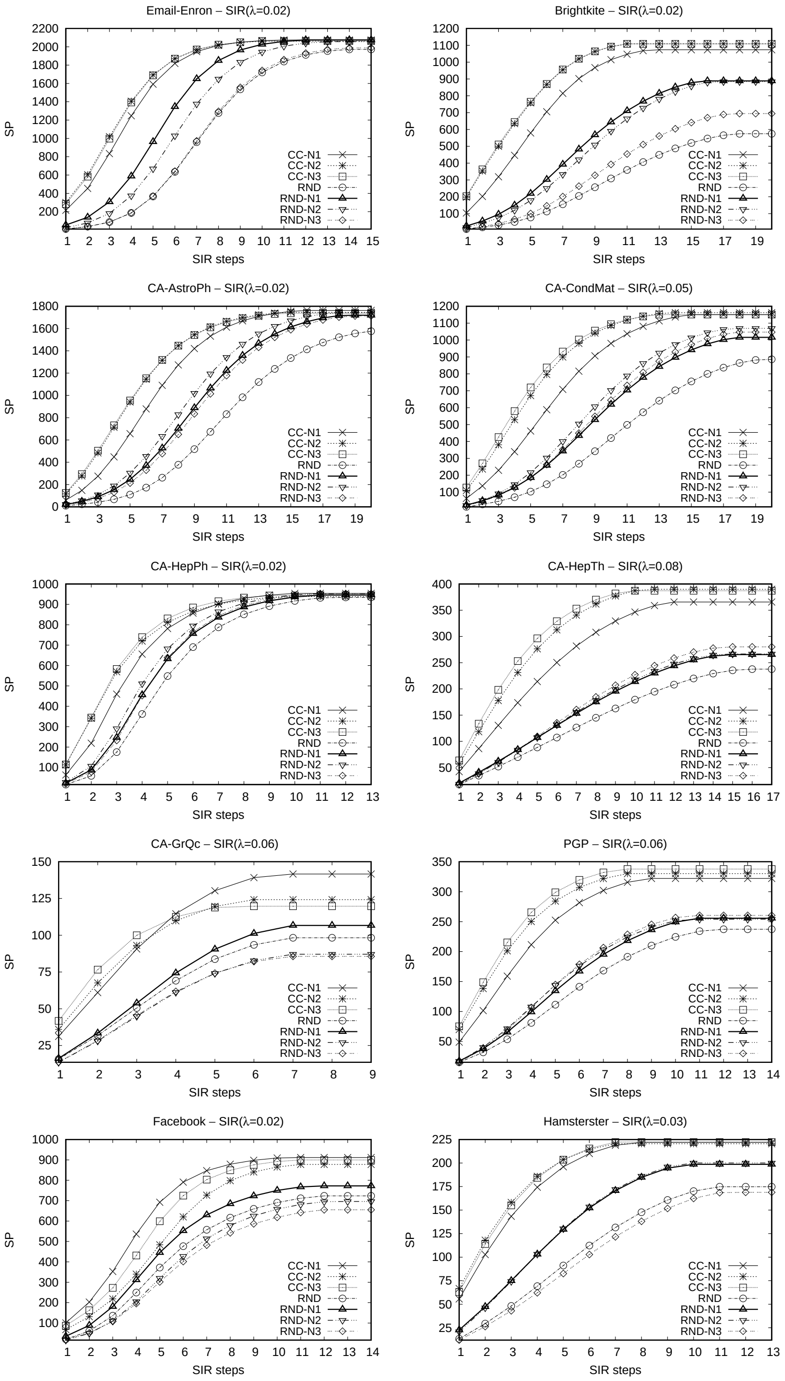
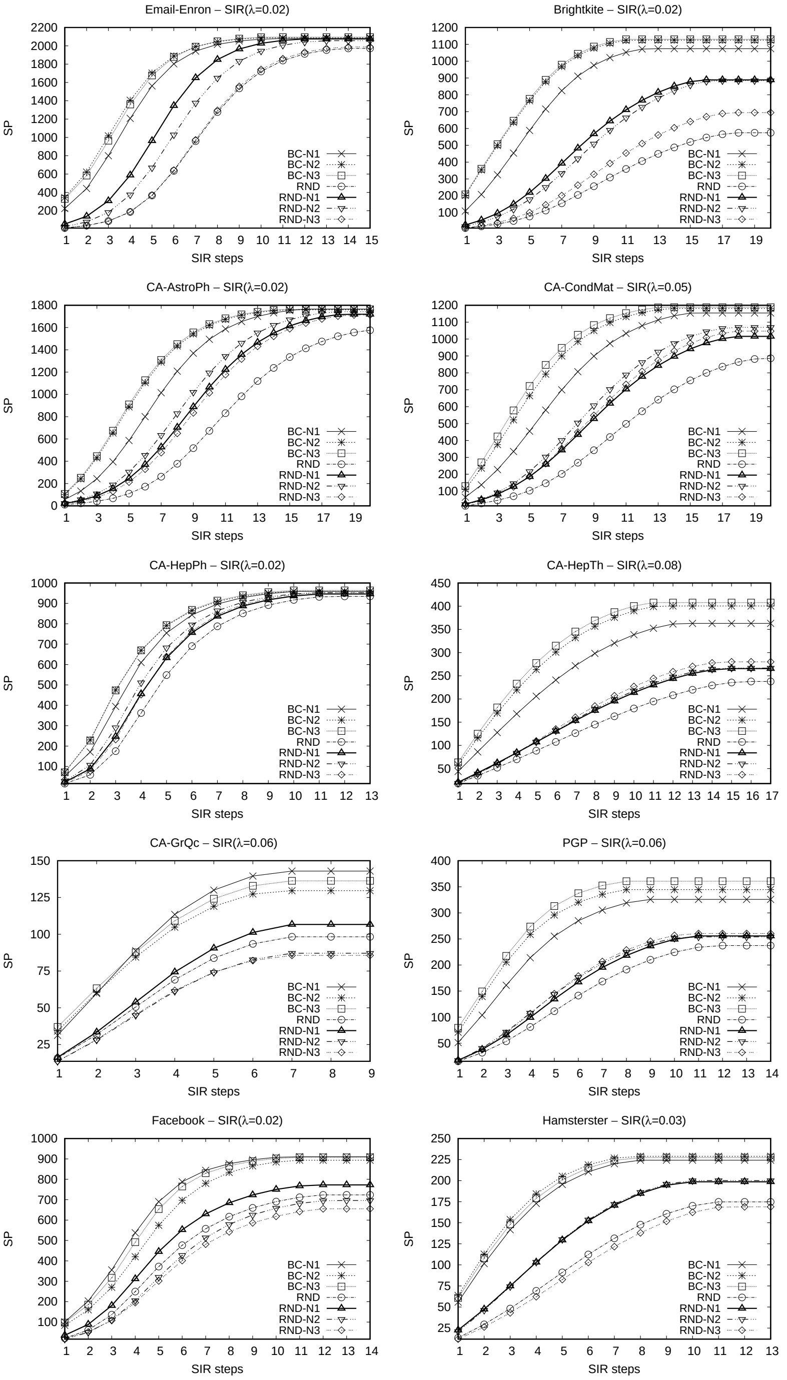


Figure C.16: Influence maximization for all networks under the SIR spreading model with cascade initiators biased towards the highest CC nodes from N_1, N_2 and N_3 of RND.


Email-Enron – SIR(λ=0.02)

Brightkite – SIR(λ=0.02)

CA-AstroPh – SIR(λ=0.02)

CA-CondMat – SIR(λ=0.05)

CA-HepPh – SIR(λ=0.02)

CA-HepTh – SIR(λ=0.08)

CA-GrQc – SIR(λ=0.06)

PGP – SIR(λ=0.06)

Facebook – SIR(λ=0.02)

Hamsterster – SIR(λ=0.03)

Figure C.17: Influence maximization for all networks under the SIR spreading model with cascade initiators biased towards the highest BC nodes from N_1, N_2 and N_3 of RND.

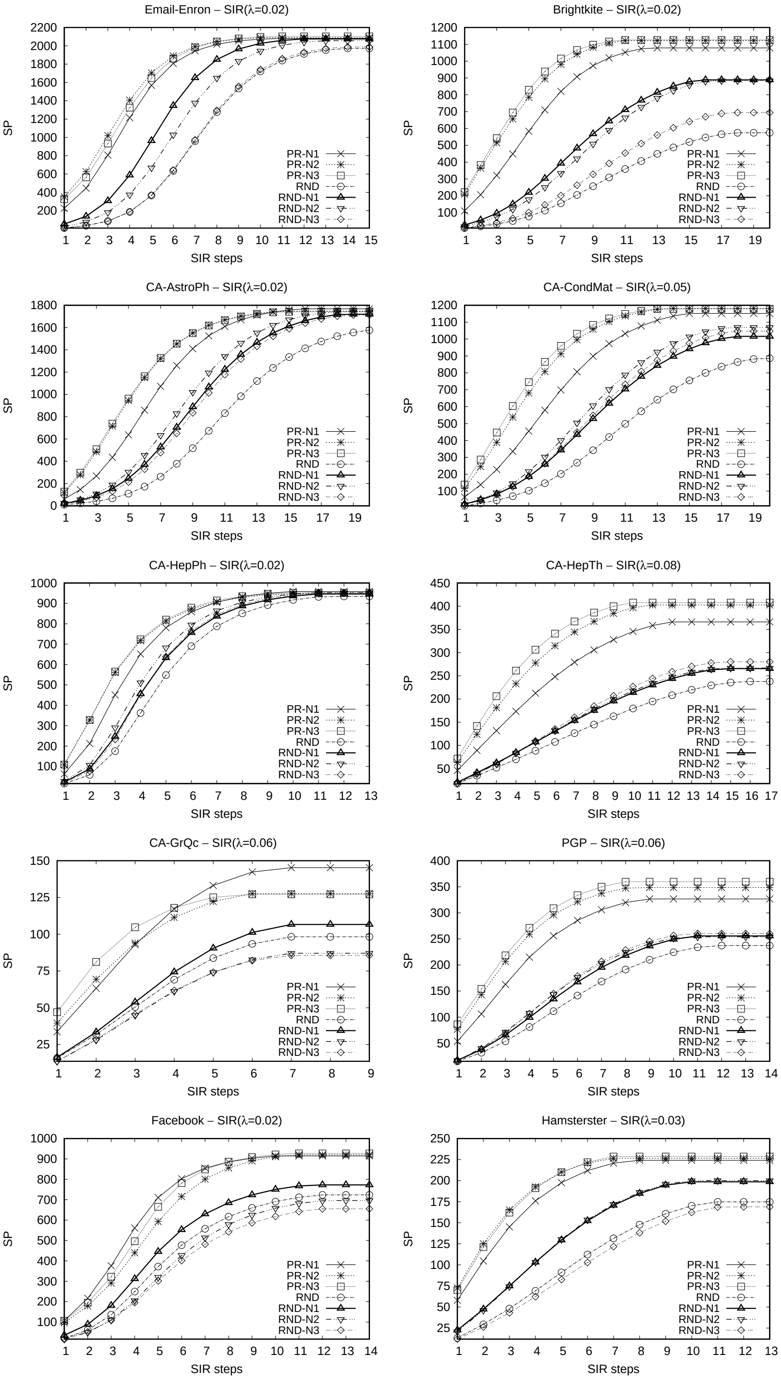


Figure C.18: Influence maximization for all networks under the SIR spreading model with cascade initiators biased towards the highest PR nodes from N_1, N_2 and N_3 of RND.

C.0.5 Detailed experiments for the spreading application of the SIS model

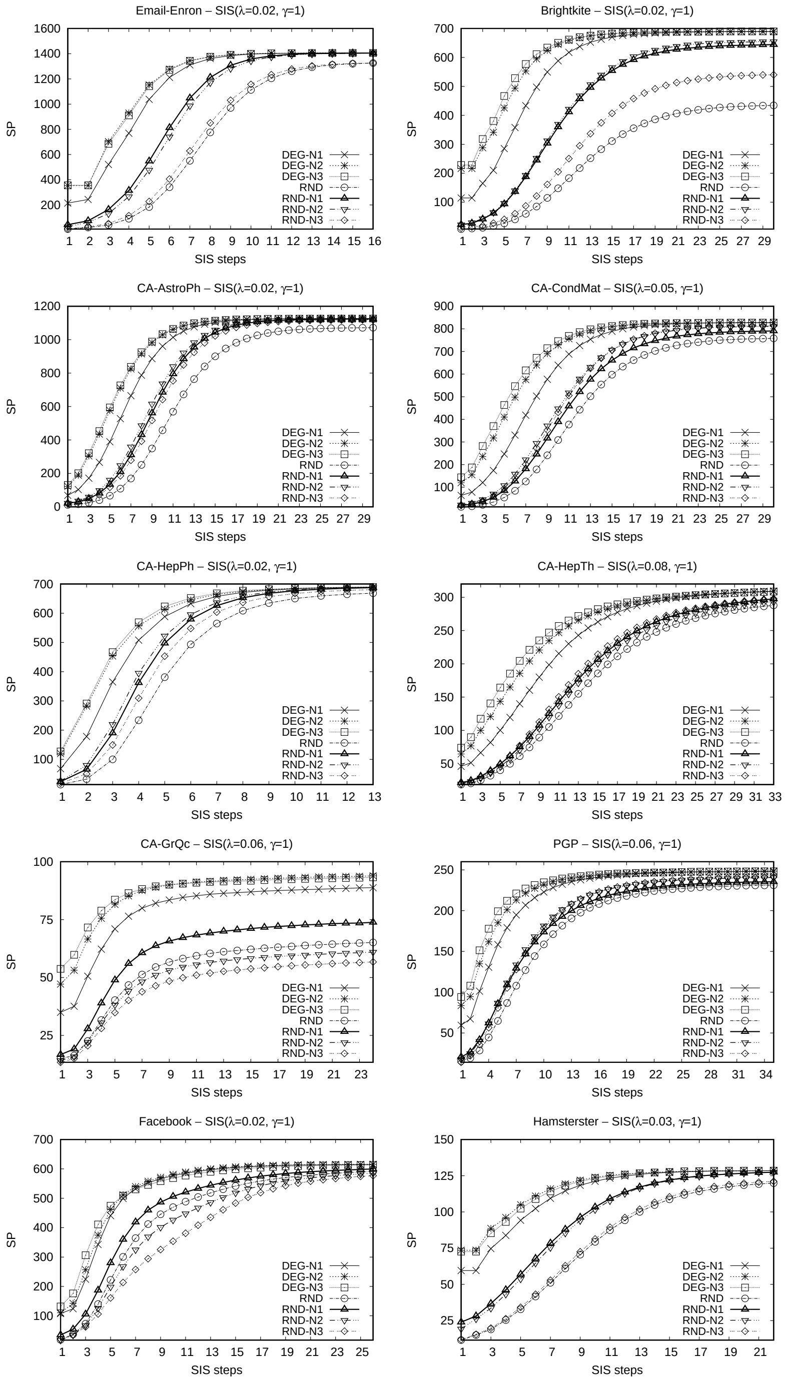


Figure C.19: Influence maximization for all networks under the SIS spreading model with cascade initiators biased towards the highest DEG nodes from N_1, N_2 and N_3 of RND.

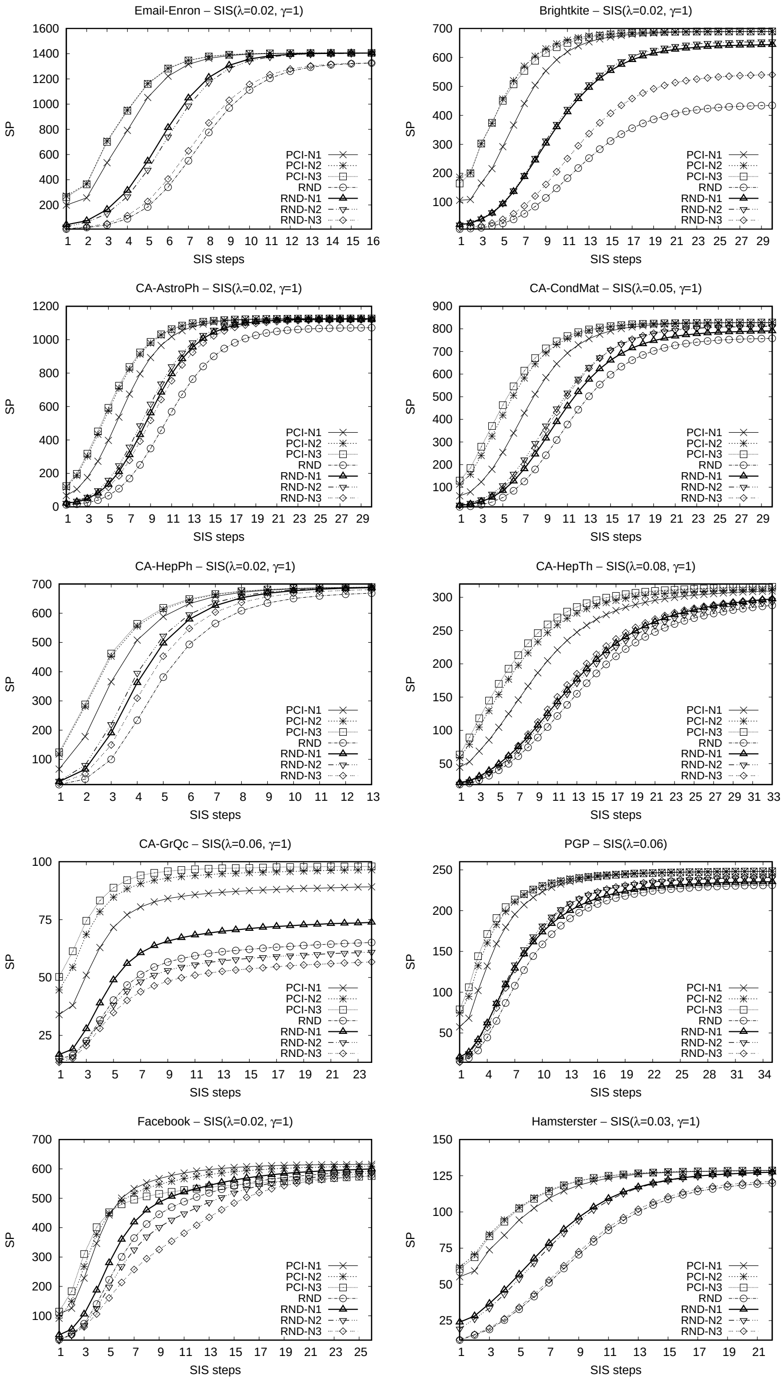


Figure C.20: Influence maximization for all networks under the SIS spreading model with cascade initiators biased towards the highest PCI nodes from N_1, N_2 and N_3 of RND.

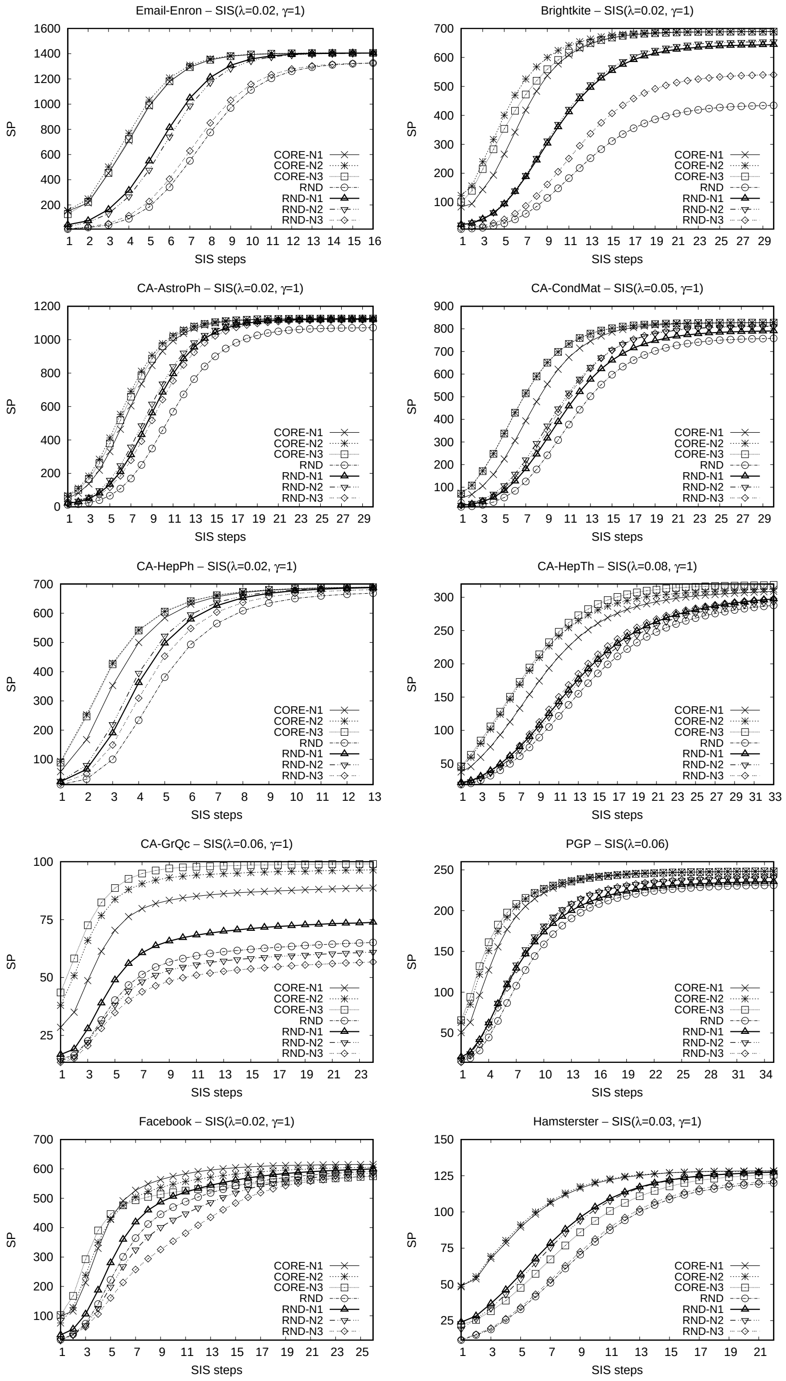


Figure C.21: Influence maximization for all networks under the SIS spreading model with cascade initiators biased towards the highest CORE nodes from N_1, N_2 and N_3 of RND.

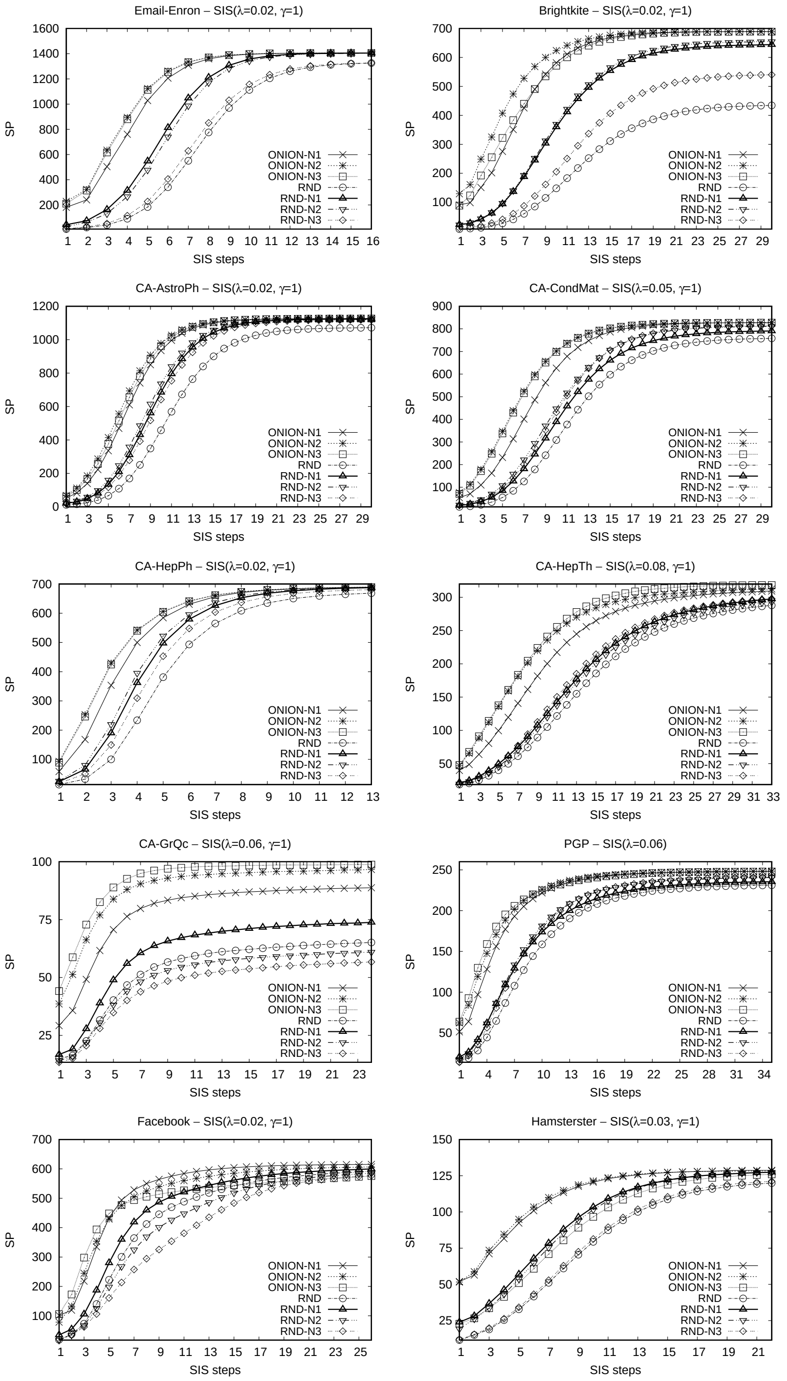


Figure C.22: Influence maximization for all networks under the SIS spreading model with cascade initiators biased towards the highest ONION nodes from N_1, N_2 and N_3 of RND.

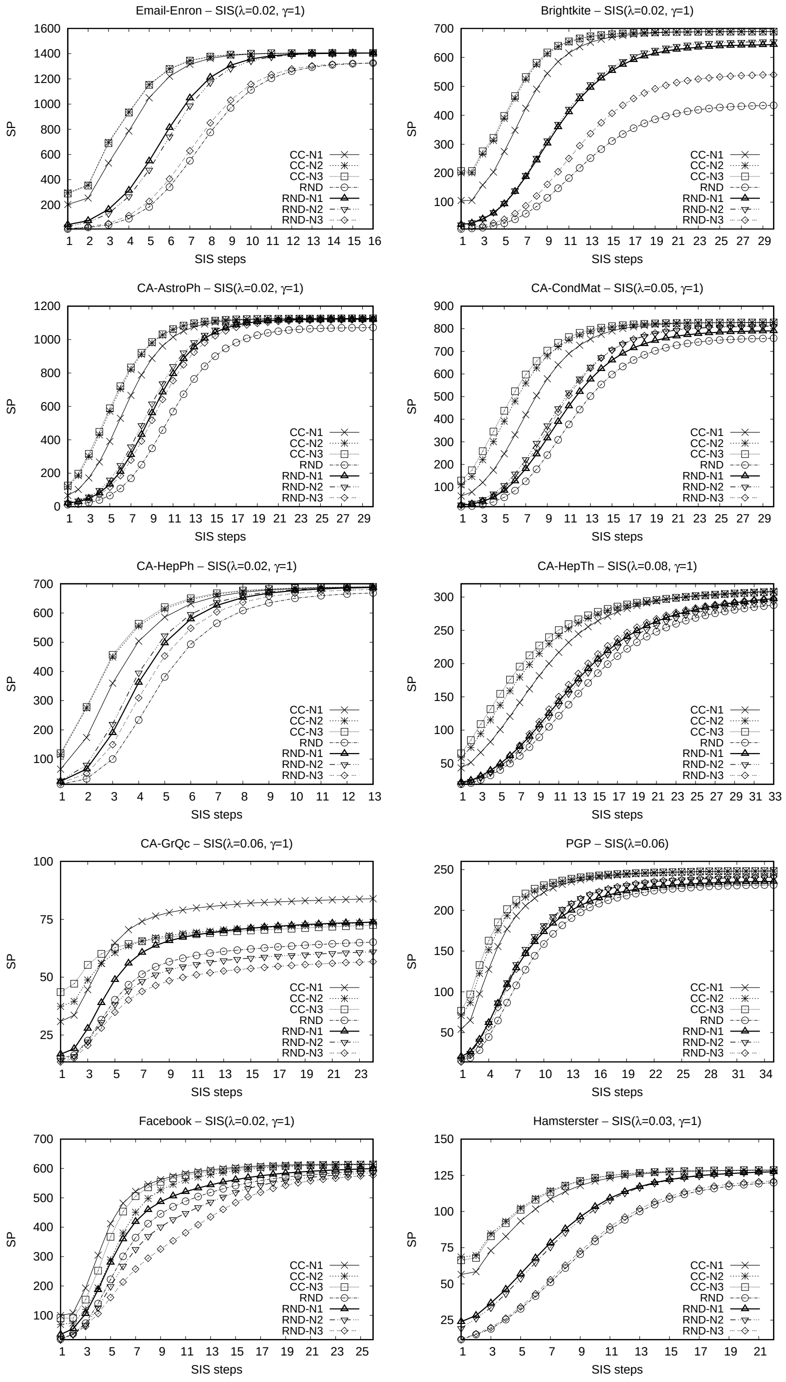


Figure C.23: Influence maximization for all networks under the SIS spreading model with cascade initiators biased towards the highest CC nodes from N_1, N_2 and N_3 of RND.

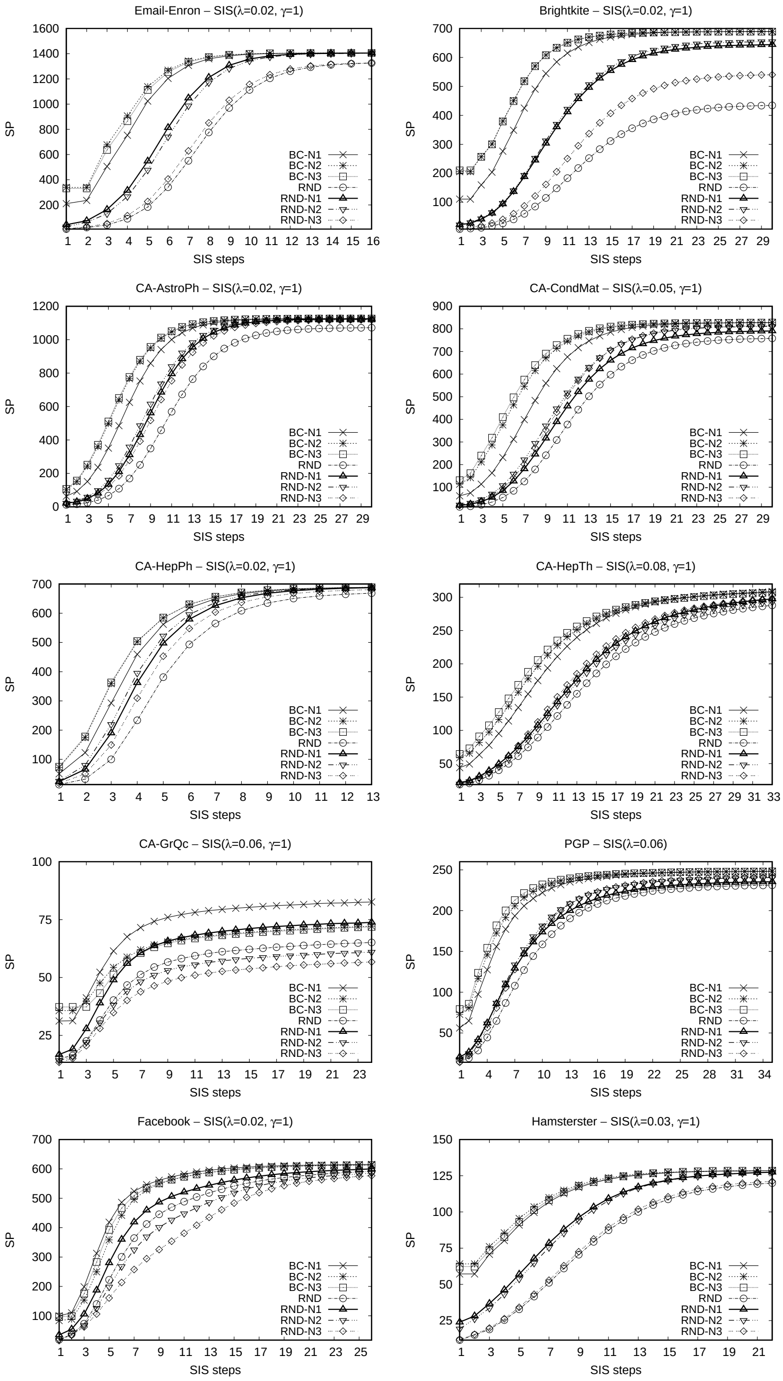


Figure C.24: Influence maximization for all networks under the SIS spreading model with cascade initiators biased towards the highest BC nodes from N_1, N_2 and N_3 of RND.

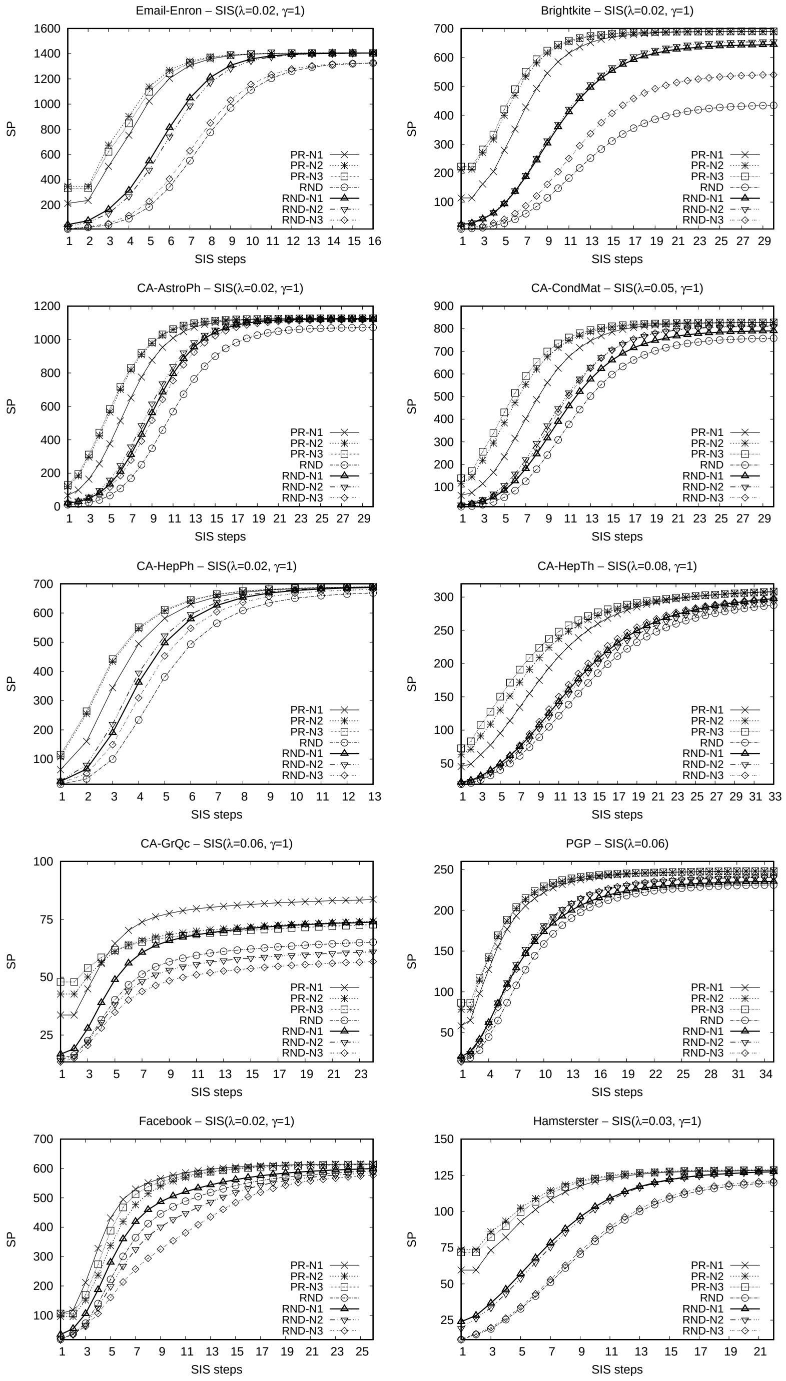


Figure C.25: Influence maximization for all networks under the SIS spreading model with cascade initiators biased towards the highest PR nodes from N_1, N_2 and N_3 of RND.

C.0.6 Detailed experiments on the spreading paradox at the individual level: SIR spreading model

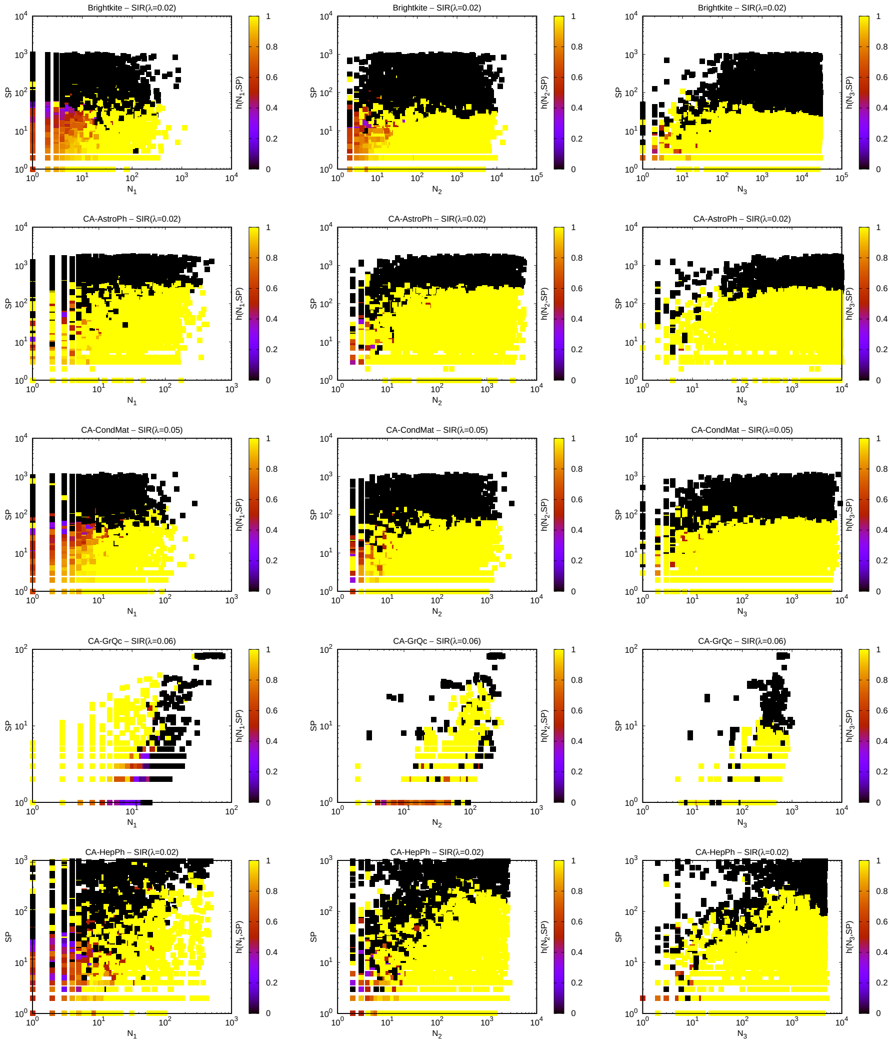
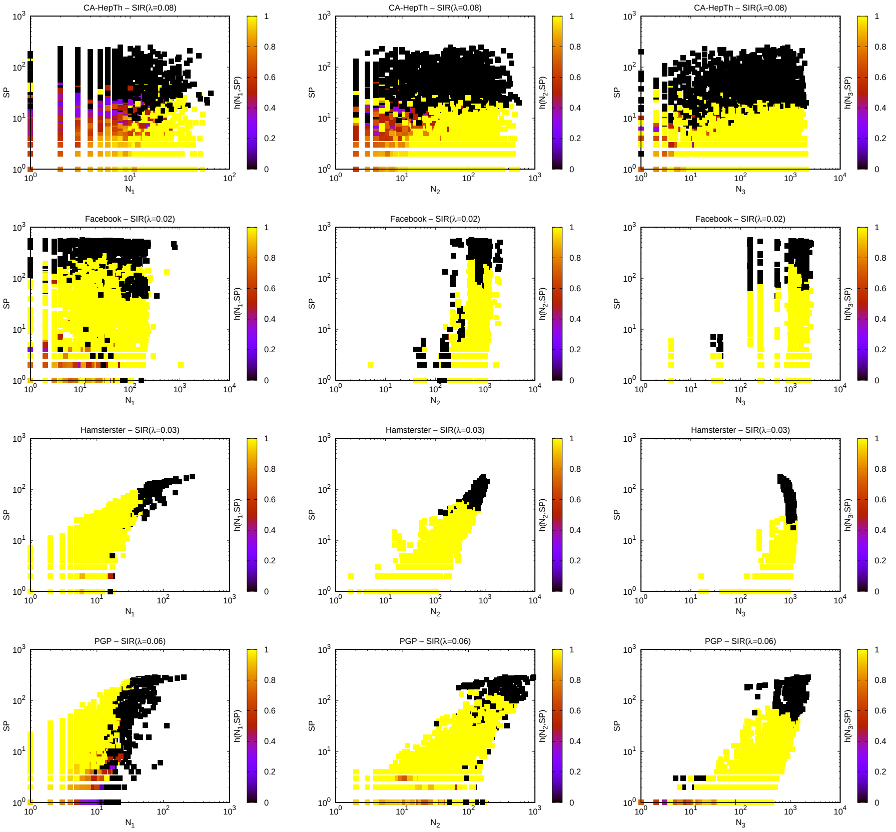


Figure C.26: Evaluation of the spreading paradox at the individual level for the SIR spreading model for the following networks: Brightkite, CA-AstroPh, CA-CondMat, CA-GrQc, CA-HepPh.


CA-HepTh – SIR(λ=0.08)

CA-HepTh – SIR(λ=0.08)

CA-HepTh – SIR(λ=0.08)

Facebook – SIR(λ=0.02)

Facebook – SIR(λ=0.02)

Facebook – SIR(λ=0.02)

Hamsterster – SIR(λ=0.03)

Hamsterster – SIR(λ=0.03)

Hamsterster – SIR(λ=0.03)

PGP – SIR(λ=0.06)

PGP – SIR(λ=0.06)

PGP – SIR(λ=0.06)

Figure C.27: Evaluation of the spreading paradox at the individual level for the SIR spreading model for the following networks: CA-HepTh, Facebook, Hamsterster, PGP.

C.0.7 Detailed experiments on the spreading paradox at the individual level: SIS spreading model

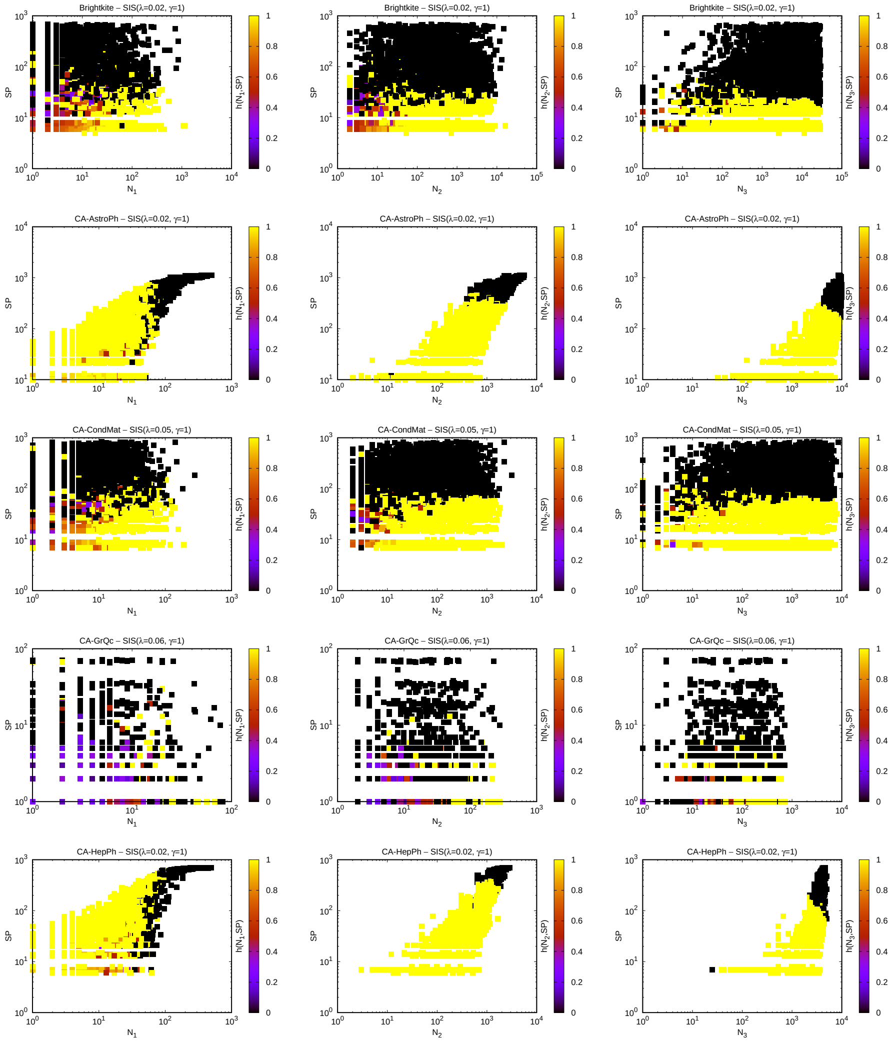


Figure C.28: Evaluation of the spreading paradox at the individual level for the SIS spreading model for the following networks: Brightkite, CA-AstroPh, CA-CondMat, CA-GrQc, CA-HepPh.

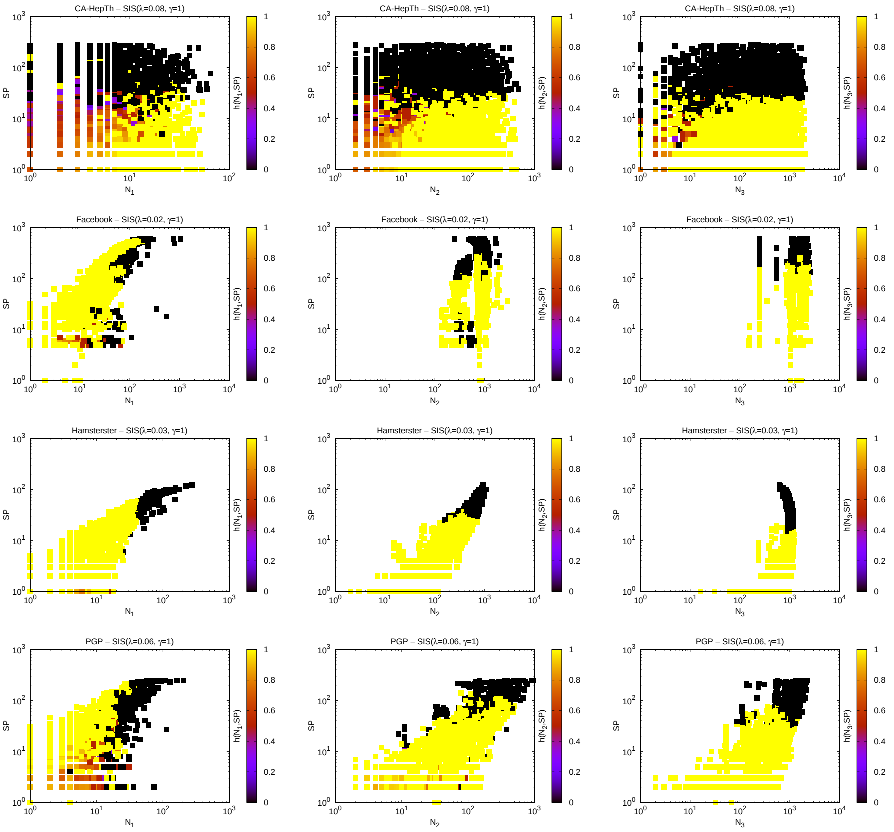


Figure C.29: Evaluation of the spreading paradox at the individual level for the SIS spreading for the following networks: CA-HepTh, Facebook, Hamsterster, PGP.

BIBLIOGRAPHY

[1] http://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/.

[2] http://www.computerworld.com/article/2960802/security/tesla-patches-model-s-after-researchers-hack-cars-software.html.

[3] http://www.symantec.com/security_response/publications/threatreport.jsp.

[4] http://www.detroitnews.com/story/business/autos/2015/02/08/report-cars-vulnerable-wireless-hacking/23094215/.

[5] http://www.techhive.com/article/221873/With_Hacking_Music_Can_Take_Control_of_Your_Car.html.

[6] T. Jungblut, Retrieved on June 4th, 2017. Available at http://codingwiththomas.blogspot.de/2011/04/graph-exploration-with-hadoop-mapreduce.html.

[7] Vertica, http://www.vertica.com/2011/09/21/counting-triangles/.

[8] WeST, *The Koblenz Network Collection*, Available at http://konect.uni-koblenz.de/.

[9] M. Bakratsas, P. Basaras, D. Katsaros, and L. Tassiulas, “Hadoop MapReduce performance on SSDs: The case of complex network analysis tasks”, in *INNS Conference on Big Data*, ser. Advances in Intelligent Systems and Computing, P. Angelov, Y. Manolopoulos, L. Iliadis, A. Roy, and M. Vellasco, Eds., vol. 529, Springer, 2017, pp. 111–119.

[10] P. Basaras, G. Iosifidis, D. Katsaros, and L. Tassiulas, “Identifying influential spreaders in complex multilayer networks: A centrality perspective”, *IEEE Transactions on Network Science and Engineering*, 2017.

[11] T. R. Foundation, *The R Project for statistical computing*, Available at https://www.r-project.org/, 2017.

[12] L. G. S. Jeub, M. W. Mahoney, P. J. Mucha, and M. A. Porter, “A local perspective on community structure in multilayer networks”, *Network Science*, 2017, Accepted. https://doi.org/10.1017/nws.2016.22.

[13] R. Pastor-Satorras and C. Castellano, “Topological structure and the H index in complex networks”, *Physical Review E*, vol. 95, no. 2, p. 022 301, 2017.

[14] Z. Zong, R. Ge, and Q. Gu, “Marcher: A heterogeneous system supporting energy-aware high performance computing and big data analytics”, *Big Data Research*, vol. 8, pp. 27–38, 2017.

[15] A.-L. Barabasi, *Network Science*. Cambridge University Press, 2016.

[16] J. Bollen, B. Goncalves, I. van de Leemput, and G. Ruan, *The happiness paradox: Your friends are happier than you*, Available at https://arxiv.org/abs/1602.02665, 2016.

[17] M. A. Al-garadi, K. D. Varathan, S. D. Ravana, E. Ahmed, and V. Chang, “Identifying the influential spreaders in multilayer interactions of online social networks”, *Journal of Intelligent & Fuzzy Systems*, vol. 31, no. 5, pp. 2721–2735, 2016.

[18] L. Hebert-Dufresne, J. A. Grochow, and A. Allard, “Multi-scale structure and topological anomaly detection via a new network statistic: The onion decomposition”, *Scientific Reports*, vol. 6, p. 31 708, 2016.

[19] J. Hong, L. Li, C. Han, B. Jin, Q. Yang, and Z. Yang, “Optimizing Hadoop framework for solid state drives”, in *Proceedings of the IEEE International Congress on Big Data*, 2016.

[20] K. R. Krish, B. Wadhwa, M. S. Iqbal, M. M. Rafique, and A. A. Butt, “On efficient hierarchical storage for big data processing”, in *Proceedings of the IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing*, 2016, pp. 403–408.

[21] S. Lee, H. Min, and S. Yoon, “Will solid-state drives accelerate your bioinformatics? In-depth profiling, performance analysis and beyond”, *Briefings in Bioinformatics*, vol. 17, no. 4, pp. 713–727, 2016.

[22] Y.-S. Lee, L. C. Quero, S.-H. Kim, J.-S. Kim, and S. Maeng, “ActiveSort: Efficient external sorting using active SSDs in the MapReduce framework”, *Future Generation Computer Systems*, vol. 65, no. C, pp. 76–89, 2016.

[23] L. Lü, T. Zhou, Q.-M. Zhang, and H. E. Stanley, “The H-index of a network node and its relation to degree and coreness”, *Nature Communications*, vol. 7, p. 10 168, 2016.

[24] N. Momeni and M. Rabbat, “Qualities and inequalities in online social networks through the lens of the generalized friendship paradox”, *PLOS One*, vol. 11, no. 2, 2016.

[25] Y. Zhuang and O. Yagan, “Information propagation in clustered multilayer networks”, *IEEE Transactions on Network Science and Engineering*, vol. 3, no. 4, pp. 211–224, 2016.

[26] S. Ahn and S. Park, “An analytical approach to evaluation of SSD effects under MapReduce workloads”, *Journal of Semiconductor Technology and Science*, vol. 15, no. 5, pp. 511–518, 2015.

[27] P. Basaras, D. Katsaros, and L. Tassiulas, “Dynamically blocking contagions in complex networks by cutting vital connections”, in *Proceedings of the IEEE International Conference on Communications (ICC)*, 2015, pp. 1170–1175.

[28] P. Basaras, L. A. Maglaras, D. Katsaros, and H. Janicke, “A robust eco-routing protocol against malicious data in vehicular network”, in *Proceedings of the IFIP Wireless and Mobile Networking Conference (WMNC)*, 2015.

[29] M. De Domenico, A. Solé-Ribalta, E. Omodei, S. Gómez, and A. Arenas, “Ranking in interconnected multilayer networks reveals versatile nodes”, *Nature Communications*, vol. 6, p. 6868, 2015.

[30] M. D. Domenico, M. A. Porter, and A. Arenas, “MuxViz: A tool for multilayer analysis and visualization of networks”, *Journal of Complex Networks*, vol. 3, no. 2, pp. 159–176, 2015.

[31] B. Fotouhi, N. Momeni, and M. Rabbat, “Generalized friendship paradox: An analytical approach”, in *Social Informatics*, ser. Lecture Notes in Computer Science, vol. 8852, 2015, pp. 339–352.

[32] M. T. Garip, M. E. Gursoy, P. Reiher, and M. Gerla, “Congestion Attacks to Autonomous Cars Using Vehicular Botnets”, in *NDSS Workshop on Security of Emerging Networking Technologies (SENT)*, San Diego, CA, Feb. 2015.

[33] U. Khan, S. Agrawal, and S. Silakari, “A detailed survey on misbehavior node detection techniques in vehicular ad hoc networks”, in *Information Systems Design and Intelligent Applications*, Springer, 2015, pp. 11–19.

[34] W. Knight, “Rebooting the automobile”, *MIT Technology Review*, vol. 118, no. 4, pp. 54–59, 2015.

[35] C. J. Kuhlman, G. Tuli, S. Swarup, M. V. Marathe, and S. S. Ravi, “Inhibiting diffusion of complex contagions in social networks: Theoretical and experimental results”, *Data Mining and Knowledge Discovery*, vol. 29, no. 2, pp. 423–465, 2015.

[36] Y. Liu, T. Tang, T. Zhou, and Y. Do, “Improving the accuracy of the k-shell method by removing redundant links: From a perspective of spreading dynamics”, *Scientific Reports*, vol. 5, p. 13 172, 2015.

[37] L. A. Maglaras, “A novel distributed intrusion detection system for vehicular ad hoc networks”, *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 4, pp. 101–106, 2015.

[38] X. Meng, “Centrality measures in multilayer networks”, University of Oxford, Tech. Rep., 2015.

[39] E. Mojahedi and M. A. Azgomi, “Modeling the propagation of topology-aware p2p worms considering temporal parameters”, *Peer-to-Peer Networking and Applications*, vol. 8, no. 1, pp. 171–180, 2015.

[40] S. Moon, J. Lee, X. Sun, and Y.-S. Kee, “Optimizing the Hadoop MapReduce framework with high-performance storage devices”, *The Journal of Supercomputing*, vol. 71, no. 9, pp. 3525–3548, 2015.

[41] F. Morone and H. A. Makse, “Influence maximization in complex networks through optimal percolation”, *Nature*, vol. 524, pp. 65–68, 2015.

[42] C. Nowzari, V. M. Preciado, and G. J. Pappas, “Analysis and control of epidemics: A survey of spreading processes on complex network”, Tech. Rep., 2015, Available at: http://arxiv.org/abs/1505.00768.

[43] J. Petit and S. E. Shladover, “Potential cyberattacks on automated vehicles”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 546–556, 2015.

[44] O. Punal, C. Pereira, A. Aguiar, and J. Gross, “Experimental characterization and modeling of rf jamming attacks on vanets”, *IEEE Transactions on Vehicular Technology*, vol. 64, no. 2, pp. 524–540, Feb. 2015.

[45] M. Salehi, R. Sharma, M. Marzolla, M. Magnani, P. Siyari, and D. Montesi, “Spreading processes in multilayer networks”, *IEEE Transactions on Network Science and Engineering*, vol. 2, no. 2, pp. 65–83, 2015.

[46] E. M. Shahrivar and S. Sundaram, “The strategic formation of multi-layer networks”, *IEEE Transactions on Network Science and Engineering*, vol. 2, no. 4, pp. 164–178, 2015.

[47] T. White, *Hadoop: The Definitive Guide*. O’Reilly Media, 2015.

[48] N. Azimi-Tafreshi, J. Gomez-Gardenes, and S. N. Dorogovtsev, “k-core percolation on multiplex networks”, *Physical Review E*, vol. 90, no. 3, p. 032 816, 2014.

[49] S. Boccaletti, G. Bianconi, R. Criado, C. I. del Genio, J. Gomez-Gardenes, M. Romance, I. Sendina-Nadal, Z. Wang, and M. Zanin, “The structure and dynamics of multilayer networks”, *Physics Reports*, vol. 544, pp. 1–222, 2014.

[50] Z. Dawei, L. Lixiang, L. Shudong, H. Yujia, and Y. Yixian, “Identifying influential spreaders in interconnected networks”, *Physica Scripta*, vol. 89, no. 1, p. 015 203, 2014.

[51] P. Devi, A. Gupta, and A. Dixit, “Comparative study of HITS and PageRank link based ranking algorithms”, *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 3, no. 2, pp. 5749–5754, 2014.

[52] Y.-H. Eom and H.-H. Jo, “Generalized friendship paradox in complex networks: The case of scientific collaboration”, *Scientific Reports*, vol. 4, p. 4603, 2014.

[53] V. Gemmetto and C. Barrat A. Cattuto, “Mitigation of infectious disease at school: Targeted class closure vs. school closure”, *BMC Infectious Diseases*, vol. 14, 694:1–694:10, 2014.

[54] P. Holme, “Analyzing temporal networks in social media”, *Proceedings of the IEEE*, vol. 102, no. 12, pp. 1922–1933, 2014.

[55] H. V. Jagadish, J. Gehrke, A. Labrinidis, Y. Papakonstantinou, J. M. Patel, R. Ramakrishnan, and C. Shahabi, “Big data and its technical challenges”, *Communications of the ACM*, vol. 57, no. 7, pp. 86–94, 2014.

[56] H.-H. Jo and Y.-H. Eom, “Generalized friendship paradox in networks with tunable degree-attribute correlation”, *Physical Review E*, vol. 90, p. 2, 2014.

[57] S. Joerer, M. Segata, B. Bloessl, R. Lo Cigno, C. Sommer, and F. Dressler, “A vehicular networking perspective on estimating vehicle collision probability at intersections”, *Vehicular Technology, IEEE Transactions on*, vol. 63, no. 4, pp. 1802–1812, 2014.

[58] B. Joonhyun and K. Sangwook, “Identifying and ranking influential spreaders in complex networks by neighborhood coreness”, *Physica A: Statistical Mechanics and its Applications*, vol. 395, no. 1, pp. 549–559, 2014.

[59] A. Kaitoua, H. Hajj, M. A. R. Saghir, H. Artail, H. Akkary, M. Awad, M. Sharafeddine, and K. Mershad, “Hadoop extensions for distributed computing on reconfigurable active SSD clusters”, *ACM Transactions on Architecture and Code Optimization*, vol. 11, no. 2, 2014.

[60] K. Kambatla and Y. Chen, “The truth about MapReduce performance on SSDs”, in *Proceedings of the USENIX Large Installation System Administration Conference (LISA)*, 2014, pp. 109–117.

[61] M. Kivela, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, “Multilayer networks”, *Journal of Complex Networks*, vol. 2, pp. 203–271, 2014.

[62] T. G. Kolda, A. Pinar, T. Plantenga, C. Seshadhri, and C. Task, “Counting triangles in massive graphs with MapReduce”, *SIAM Journal on Scientific Computing*, vol. 36, no. 5, pp. 48–77, 2014.

[63] K. R. Krish, M. S. Iqbal, and A. R. Butt, “VENU: Orchestrating SSDs in Hadoop storage”, in *Proceedings of the IEEE International Conference on Big Data (BigData)*, 2014, pp. 207–212.

[64] A. N. Langville and C. D. Meyer, *Who’s #1?: The Science of Rating and Ranking*. Princeton University Press, 2014.

[65] J. Leskovec and A. Krevl, *SNAP datasets: Stanford large network dataset collection*, `http://snap.stanford.edu/data`, 2014.

[66] Q. Li, T. Zhou, L. Lv, and D. Chen, “Identifying influential spreaders by Weighted LeaderRank”, *Physica A: Statistical Mechanics and its Applications*, vol. 404, pp. 47–55, 2014.

[67] M. Milojevic and V. Rakocevic, “Distributed road traffic congestion quantification using cooperative VANETs”, in *13th IFIP/IEEE Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, Jun. 2014, pp. 203–210.

[68] S. Moon, J. Lee, and Y. S. Kee, “Introducing SSDs to the Hadoop MapReduce framework”, in *Proceeding of the IEEE International Conference on Cloud Computing (CLOUD)*, 2014, pp. 272–279.

[69] K. Pechlivanidou, D. Katsaros, and L. Tassiulas, “MapReduce-based distributed k-shell decomposition for online social networks”, in *Proceedings of the International Workshop on Personalized Web Tasking (PWT)*, 2014, pp. 30–37.

[70] S. Pei, L. Muchnik, J. A. Andrade, Z. Zheng, and H. A. Makse, “Searching for superspreaders of information in real-world social media”, *Nature Scientific Reports*, vol. 4, p. 5547, 2014.

[71] P. Saxena and D. Chou, “How much solid state drive can improve the performance of Hadoop cluster? Performance evaluation of Hadoop on SSD and HDD”, *International Journal of Modern Communication Technologies & Research*, vol. 2, no. 5, 2014.

[72] K. Scaman, A. Kalogeratos, and N. Vayatis, “Dynamic treatment allocation for epidemic control in arbitrary networks”, in *Proceedings of the ACM WSDM Workshop on the Diffusion Networks and Cascade Analytics (DiffNet)*, 2014.

[73] A. Solé-Ribalta, M. De Domenico, S. Gómez, and A. Arenas, “Centrality rankings in multiplex networks”, in *Proceedings of the ACM Conference on Web Science*, 2014, pp. 149–155.

[74] A. M. d. Souza, A. Boukerche, G. Maia, R. I. Meneguette, A. A. Loureiro, and L. A. Villas, “Decreasing Greenhouse Emissions Through an Intelligent Traffic Information System Based on Inter-vehicle Communication”, in *12th International Symposium on Mobility Management and Wireless Access (MobiWac)*, ACM, 2014, pp. 91–98, ISBN: 978-1-4503-3026-8. DOI: `10.1145/2642668.2642677`.

[75] A. Tagarelli and R. Interdonato, “Understanding lurking behaviors in social networks across time”, in *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2014, pp. 51–55.

[76] K. Zhang and X.-W. Chen, “Large-scale deep belief nets with MapReduce”, *IEEE Access*, vol. 2, pp. 395–403, 2014.

[77] P. Basaras, D. Katsaros, and L. Tassiulas, “Detecting influential spreaders in complex, dynamic networks”, *IEEE Computer magazine*, vol. 46, no. 4, pp. 26–31, 2013.

[78] M. G. Campiteli, A. J. Holanda, L. D. H. Soares, P. R. C. Soles, and O. Kinouchi, “Lobby index as a network centrality measure”, *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 21, pp. 5511–5515, 2013.

[79] A. Cardillo, J. Gomez-Gardenes, M. Zanin, M. Romance, D. Papo, F. del Pozo, and S. Boccaletti, “Emergence of network features from multiplexity”, *Nature Scientific Reports*, vol. 3, p. 1344, 2013.

[80] Y. A. Daraghmi, I. Stojmenovic, and C. W. Yi, “A taxonomy of data communication protocols for vehicular ad hoc networks”, *Mobile Ad Hoc Networking: Cutting Edge Directions*, pp. 517–544, 2013.

[81] Y. A. Daraghmi, C. W. Yi, I. Stojmenovic, and K. Abdulaziz, "Forwarding methods in data dissemination and routing protocols for vehicular ad hoc networks", *IEEE Network*, vol. 27, pp. 74–79, 2013.

[82] M. D. Domenico, A. Sole-Ribalta, E. Cozzo, M. Kivela, Y. Moreno, M. A. Porter, S. Gomez, and A. Arenas, "Mathematical formulation of multilayer networks", *Physical Review X*, vol. 3, p. 041 022, 2013.

[83] M. Eidsaa and E. Almaas, "s-core network decomposition: A generalization of k-core analysis to weighted networks", *Physical Review E*, vol. 88, 062819:1–062819:6, 2013.

[84] L. Fan, Z. Lu, W. Wu, B. Thuraisingham, H. Ma, and B. Y., "Least cost rumor blocking in social networks", in *Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2013, pp. 540–549.

[85] K. Gong, M. Tang, P. M. Hui, H. F. Zhang, D. Younghae, and Y. C. Lai, "An efficient immunization strategy for community networks", *PLoS ONE*, vol. 8, no. 12, 2013.

[86] A. Goyal, F. Bonchi, L. Lakshmanan, and S. Venkatasubramanian, "On minimizing budget and time in influence propagation over social networks", *Social Network Analysis and Mining*, vol. 3, no. 2, pp. 179–192, 2013.

[87] A. Halu, R. J. Mondragón, P. Panzarasa, and G. Bianconi, "Multiplex PageRank", *PLOS One*, vol. 8, no. 10, e78293, 2013.

[88] L. Hebert-Dufresne, A. Allard, J. G. Young, and L. J. Dube, "Global efficiency of local immunization on complex networks", *Nature Scientific Reports*, vol. 3, 2013.

[89] N. O. Hodas, F. Kooti, and K. Lerman, "Friendship paradox redux: Your friends are more interesting than you", in *Proceedings of the International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2013.

[90] J. B. Holthoefer, S. Meloni, B. Goncalves, and Y. Moreno, "Emergence of influential spreaders in modified rumor models", *Journal of Statistical Physics*, vol. 151, pp. 383–393, 2013.

[91] B. H. Javier, R. A. Banos, B. S. Gonzalez, and Y. Moreno, "Cascading behaviour in complex socio-technical networks", *Journal of Complex Networks*, vol. 1, pp. 3–24, 2013.

[92] S.-H. Kang, D.-H. Koo, W.-H. Kang, and S.-W. Lee, "A case for flash memory SSD in Hadoop applications", *International Journal of Control and Automation*, vol. 6, no. 1, pp. 201–210, 2013.

[93] N. Kourtellis, T. Alahakoon, R. Simha, A. Iamnitchi, and R. Tripathi, "Identifying high betweenness centrality nodes in large social networks", *Social Network Analysis and Mining*, vol. 3, no. 4, pp. 899–914, 2013.

[94] R. Krikorian, *New tweets per second record, and how!*, Twitter Official Blog. August 16, 2013.

[95] C. J. Kuhlman, G. Tuli, S. Swarup, M. V. Marathe, and S. S. Ravi, “Blocking simple and complex contagion by edge removal”, in *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2013, pp. 399–408.

[96] C.-T. Li, T.-T. Kuo, C.-T. Ho, S.-C. Hong, W.-S. Lin, and S.-D. Lin, “Modeling and evaluating information propagation in a microblogging social network”, *Social Network Analysis and Mining*, vol. 3, no. 3, pp. 341–357, 2013.

[97] J.-G. Liu, Z.-M. Ren, and Q. Guo, “Ranking the spreading influence in complex networks”, *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 18, pp. 4154–4159, 2013.

[98] L. A. Maglaras, P. Basaras, and D. Katsaros, “Exploiting vehicular communications for reducing co2 emissions in urban environments”, in *Connected Vehicles and Expo (ICCVE), International Conference on*, IEEE, 2013, pp. 32–37.

[99] N. P. Nguyen, G. Yan, and M. T. Thai, “Analysis of misinformation containment in online social networks”, *Computer Networks*, vol. 57, no. 10, pp. 2133–2146, 2013.

[100] R. Sumbaly, J. Kreps, and S. Shah, “The big data ecosystem at LinkedIn”, in *Proceedings of the ACM SIGMOD International Conference on the Management of Data (SIGMOD)*, 2013, pp. 1125–1134.

[101] O. Trullols-Cruces, M. Fiore, and J. M. Barcelo-Ordinas, “Understanding, modeling and taming mobile malware epidemics in a large-scale vehicular network”, in *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM)*, 2013.

[102] P. Wang, G. Robins, P. Pattison, and E. Lazega, “Exponential random graph models for multilevel networks”, *Social Networks*, vol. 35, pp. 96–115, 2013.

[103] X. Wei, N. C. Valler, B. A. Prakash, I. Neamtiu, M. Faloutsos, and C. Faloutsos, “Competing memes propagation on networks: A network science perspective”, *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 6, pp. 1049–1060, 2013.

[104] D. Wu, W. Xie, X. Ji, W. Luo, J. He, and D. Wu, “Understanding the impacts of solid-state storage on the Hadoop performance”, in *Proceedings of the International Conference on Advanced Coud and Big Data*, 2013, pp. 125–130.

[105] A. Zeng and C. J. Zhang, “Ranking spreaders by decomposing complex networks”, *Physics Letters A*, vol. 377, no. 14, pp. 1031–1035, 2013.

[106] B. Aditya Prakash, D. Chakrabarti, N. C. Valler, M. Faloutsos, and C. Faloutsos, “Threshold conditions for arbitrary cascade models on arbitrary networks”, *Knowledge and Information Systems*, vol. 33, no. 3, pp. 549–575, 2012.

[107] S. Aral and D. Walker, “Identifying influential and susceptible members of social networks”, *Science*, vol. 337, pp. 337–341, 2012.

[108] V. Bibhu, K. Roshan, K. B. Singh, and D. K. Singh, “Performance analysis of black hole attack in Vanet”, *International Journal of Computer Network and Information Security (IJCNIS)*, vol. 4, no. 11, pp. 47–54, 2012.

[109] J. Borge-Holthoefer and Y. Morebo, “Absence of influential spreaders in rumor dynamics”, *Physical Rev. E*, vol. 85, no. 2, 2012.

[110] D. Chen, L. Lu, M. S. Shang, Y. C. Zhang, and T. Zhou, “Identifying influential nodes in complex networks”, *Physica A: Statistical Mechanics and its Applications*, vol. 391, no. 4, pp. 1777–1787, 2012.

[111] B. Doer, F. Mahmoud, and T. Friedrich, “Why rumors spread so quickly in social networks”, *First Monday*, vol. 55, no. 6, pp. 70–75, 2012.

[112] B. Han and A. Srinivasan, “Your friends have more friends than you do: Identifying influential mobile users through random walks”, in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, 2012, pp. 5–14.

[113] J. B. Holthoefer, A. Rivero, and Y. Moreno, “Locating privileged spreaders on an online social network”, *Physical Review E*, vol. 85, 066123:1–066123:6, 2012.

[114] N. Islam, M. Rahman, J. Jose, R. Rajachandrasekar, H. Wang, H. Subramoni, C. Murthy, and D. Panda, “High performance RDMA-design of HDFS over InfiniBand”, in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis (SC)*, 2012.

[115] C.-H. Kim and I.-H. Bae, “A misbehavior-based reputation management system for vanets”, in *Embedded and Multimedia Computing Technology and Service*, Springer, 2012, pp. 441–450.

[116] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabasi, “Control centrality and hierarhical structure in complex networks”, *PLOS One*, vol. 7, no. 9, 2012.

[117] C. Min, K. Kim, H. Cho, S.-W. Lee, and Y. Eom, “SFS: Random write considered harmful in solid state drives”, in *Proceedings of the USENIX Conference on File and Storage Technologies (FAST)*, 2012.

[118] A. Saumell-Mendiola, M. A. Serrano, and M. Boguñá, “Epidemic spreading on interconnected networks”, *Physical Review E*, vol. 86, no. 2, p. 026 106, 2012.

[119] I. Stojmenovic, A. Khan, and N. Zaguia, “Broadcasting with seamless transition from static to highly mobile wireless ad hoc, sensor and vehicular networks”, *International Journal of Parallel, Emergent and Distributed Systems*, vol. 27, pp. 225–234, 2012.

[120] H. Tong, B. A. Prakash, T. Eliassi-Rad, M. Faloutsos, and C. Faloutsos, “Gelling, and melting, large graphs by edge manipulation”, in *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*, 2012, pp. 245–254.

[121] A. M. Vegni, A. Stramacci, and E. Natalizio, “SRB: A selective reliable broadcast protocol for safety applications in vanets”, in *Proc. of Intl. Conf. on Selected Topics in Mobile & Wireless Networking*, 2012.

[122] J. Wang and J. Cheng, “Truss decomposition in massive networks”, *Proceedings of the VLDB Environment*, vol. 5, no. 9, pp. 812–823, 2012.

[123] L. Akritids, D. Katsaros, and P. Bozanis, “Identifying the productive and influential bloggers in a community”, *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews,*, vol. 41, no. 5, pp. 759–764, 2011.

[124] Y. Chen, A. Ganapathi, R. Griffith, and R. Katz, “The case for evaluating Mapreduce performance using workload suites”, in *Proceedings of the IEEE International Symposium on Modelling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, 2011, pp. 390–399.

[125] S. Huang, J. Huang, J. Dai, T. Xie, and B. Huang, “The HiBench benchmark suite: Characterization of the MapReduce-based data analysis”, in *Frontiers in Information and Software as Services*, ser. Lecture Notes in Business Information Processing, vol. 74, Springer, 2011, pp. 209–228.

[126] L. Lu, Y. C. Zhang, C. H. Yeung, and T. Zhou, “Leaders in social networks, the Delicious case”, *PLoS ONE*, vol. 6, 0021202:1–0021202:9, 2011.

[127] G. Remy, S.-M. Senouci, F. Jan, and Y. Gourhant, “Lte4v2x: LTE for a centralized vanet organization”, in *Global Telecommunications Conference (GLOBECOM), IEEE*, 2011, pp. 1–6.

[128] J. Sahoo, E. H.-K. Wu, P. K. Sahu, and M. Gerla, “Binary-partition-assisted amc-layer broadcast for emergency message dissemination in vanets”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, pp. 757–770, 2011.

[129] C. Sommer, R. German, and F. Dressler, “Bidirectionally coupled network and road traffic simulation for improved IVC analysis”, *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, 2011.

[130] I. A. Sumra, I. Ahmad, H. Hasbullah, and J.-L. bin Ab Manan, “Classes of attacks in vanet”, in *Electronics, Communications and Photonics Conference (SIECPC), Saudi International*, 2011, pp. 1–5.

[131] F. Bai, D. Stancil, and H. Krishnan, “Toward understanding characteristics of Dedicated Short Range Communications (DSRC) from a perspective of vehicular network engineers”, in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, 2010, pp. 329–340.

[132] S. V. Buldyrev, R. Parshani, P. Gerald, S. H. Eugene, and H. Shlomo, “Catastrophic cascade of failures in interdependent networks”, *Nature*, vol. 464, pp. 1025–1028, 2010.

[133] C. Castellano and R. Pastor-Satorras, “Thresholds for epidemic spreading in networks”, *Physical Review Letters*, vol. 105, no. 218701-1–218701-4, 2010.

[134] D. Caveney, “Cooperative vehicular safety applications”, *IEEE Control Systems magazine*, vol. 30, no. 4, 2010.

[135] L. Cheng and R. Shakya, “VANET worm spreading from traffic modeling”, in *Proceedings of the IEEE Radio and Wireless Symposium (RWS)*, 2010, pp. 669–672.

[136] N. A. Christakis and J. H. Fowler, “Social network sensors for early detection of contagious outbreaks”, *PLoS ONE*, vol. 5, no. 9, 2010.

[137] S. Huang, J. Huang, J. Dai, T. Xie, and B. Huang, “The HiBench benchmark suite: Characterization of the MapReduce-based data analysis”, in *Proceedings of the IEEE International Conference on Data Engineering Workshops (ICDEW)*, 2010, pp. 41–51.

[138] D. Katsaros, N. Dimokas, and L. Tassiulas, “Social network analysis concepts in the design of wireless ad hoc network protocols”, *IEEE Network magazine*, vol. 24, no. 6, pp. 23–29, 2010.

[139] M. Kitsak, L. K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H. E. Stanley, and H. A. Makse, “Identification of influential spreaders in complex networks”, *Nature Physics*, vol. 6, pp. 888–893, 2010.

[140] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage, “Experimental security analysis of a modern automobile”, in *Proceedings of the IEEE Symposium in Security and Privacy (SP)*, 2010, pp. 447–462.

[141] P. J. Mucha and M. A. Porter, “Communities in multislice voting networks”, *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 20, no. 4, p. 041 108, 2010.

[142] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J.-K. Onnela, “Community structure in time-dependent, multiscale, and multiplex networks”, *Science*, vol. 328, no. 5980, pp. 876–878, 2010.

[143] M. Muter, A. Groll, and F. C. Freiling, “A structured approach to anomaly detection for in-vehicle networks”, in *Proceedings of the IEEE International Conference on Information Assurance and Security (IAS)*, 2010, pp. 92–98.

[144] M. E. J. Newman, *Networks: An Introduction*. Oxford University Press, 2010.

[145] S. Sur, H. Wang, J. Huang, X. Ouyang, and D. Panda, “Can high-performance interconnects benefit Hadoop distributed file system”, in *Proceedings of the Workshop on Micro Architectural Support for Virtualization, Data Center Computing, and Clouds (MASVDC)*, 2010.

[146] K. Thomas and D. M. Nicol, "The Koobface botnet and the rise of social malware", in *Proceedings of the International Conference on Malicious and Unwanted Software (MALWARE)*, 2010, pp. 63–70.

[147] Y. T. Tseng, R. H. Jan, C. Chen, C. F. Wang, and H. H. Li, "A vehicle-density-based forwarding scheme for emergency message broadcasts in vanets", in *IEEE 7th International Conference on Mobile Adhoc and Sensor Systems*, 2010, pp. 703–708.

[148] K. Weil, *Measuring tweets*, Twitter Official Blog. February 22, 2010.

[149] J. Weng, E. P. Lim, J. Jang, and Q. He, "TwitterRank: Finding topic-sensitive influential Twitterers", in *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, 2010, pp. 261–270.

[150] N. Dimokas, D. Katsaros, L. Tassiulas, and Y. Manolopoulos, "High performance, low overhead cooperative caching for wireless sensor networks", in *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2009.

[151] J. Domingo-Ferrer and Q. Wu, "Safety and privacy in vehicular communications", in *Privacy in Location-Based Applications*, Springer, 2009, pp. 173–189.

[152] A. Korn, A. Schubert, and A. Telcs, "Lobby index in networks", *Physica A: Statistical Mechanics and its Applications*, vol. 388, no. 11, pp. 2221–2226, 2009.

[153] D. Lazer et al, "Computational social science", *Science 06*, vol. 323, pp. 721–723, 2009.

[154] G. Pallis, D. Katsaros, M. D. Dikaiakos, N. Loulloudes, and L. Tassiulas, "On the structure and evolution of vehicular networks", in *Proceedings of IEEE/ACM MASCOTS*, 2009, pp. 502–511.

[155] T. Smieszek, "A mechanistic model of infection: Why duration and intensity of contacts should be included in models of disease spread", *Theoretical Biology and Medical Modelling*, vol. 6, 2009.

[156] H.-F. Zhang, K.-Z. Li, X.-C. Fu, and B.-H. Wang, "An efficient control strategy of epidemic spreading on scale-free networks", *Chinese Physics Letters*, vol. 26, no. 6, 2009.

[157] W. Zhao, H. Ma, and Q. He, "Parallel k-means clustering based on MapReduce", in *Proceedings of the International Conference on Cloud Computing (CloudCom)*, 2009, pp. 674–679.

[158] Z. Cao, J. Kong, U. Lee, M. Gerla, and Z. Chen, "Proof-of-relevance: Filtering false data via authentic consensus in vehicle ad-hoc networks", in *INFOCOM Workshop*, IEEE, 2008, pp. 1–6.

[159] M. Kimura, K. Saito, and H. Motoda, "Minimizing the spread of contamination by blocking links in a network", in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, vol. 2, 2008, pp. 1175–1180.

[160] U. E. Larson, D. K. Nilsson, and E. Jonsson, “An approach to specification-based attack detection for in-vehicle networks”, in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2008, pp. 220–225.

[161] S. Lee, B. Moon, C. Park, and S. Kim, “A case for flash memory SSD in enterprise database applications”, in *Proceedings of the ACM Conference on the Management of Data (SIGMOD)*, 2008, pp. 1075–1086.

[162] V. Verendel, D. K. Nilsson, U. E. Larson, and E. Jonsson, “An approach to using honeypots in in-vehicle networks”, in *Proceedings of the IEEE Vehicular Technology Conference-Fall (VTC-Fall)*, 2008.

[163] S. Antonatos, P. Akritidis, E. P. Markatos, and K. G. Anagnostakis, “Defending against hitlist worms using network address space randomization”, *Computer Networks*, vol. 51, no. 12, pp. 3471–3490, 2007.

[164] L. Buttyan, T. Holczer, and I. Vajda, “On the effectiveness of changing pseudonyms to provide location privacy in VANETs”, in *Security and Privacy in Ad-hoc and Sensor Networks*, ser. Lecture Notes in Computer Science, vol. 4572, 2007, pp. 129–141.

[165] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. M. van Briesen, and N. S. Glance, “Cost-effective outbreak detection in networks”, in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2007, pp. 420–429.

[166] N. Litvak, W. R. W. Scheinhardt, and Y. Volkovich, “In-degree and PageRank: Why do they follow similar power laws?”, *Internet Mathematics*, vol. 4, no. 2, pp. 175–198, 2007.

[167] M. Raya, P. Papadimitratos, I. Aad, D. Jungels, and J.-P. Hubaux, “Eviction of misbehaving and faulty nodes in vehicular networks”, *Selected Areas in Communications, IEEE Journal on*, vol. 25, no. 8, pp. 1557–1568, 2007.

[168] S. Biswas, R. Tatchikou, and F. Dion, “Vehicle-to-vehicle wireless communication protocols for enhancing highway traffic safety”, *Communications Magazine, IEEE*, vol. 44, no. 1, pp. 74–82, 2006.

[169] S. Eubank, V. S. Anil-Kumar, M. Marathe, A. Srinivasan, and N. Wang, “Structure of social contact networks and their impact on epidemics”, in *AMS-DIMACS Special Issue on Epidemiology*, 2006, pp. 181–213.

[170] J. Guo and N. Balon, “Vehicular ad hoc networks and dedicated short-range communication”, *Univesity of Michigan, September*, vol. 22, 2006.

[171] A. N. Langville and C. D. Meyer, *Google’s PageRank and Beyond: The Science of Search Engine Rankings*. Princeton University Press, 2006.

[172] C.-Y. Lee, *Correlations among centrality measures in complex networks*, Available at https://arxiv.org/abs/physics/0605220, 2006.

[173] M. Nekovee, “Modeling the spread of worm epidemics in vehicular ad hoc networks”, in *Proceedings of the IEEE Vehicular Technology Conference-Spring (VTC-Spring)*, 2006, pp. 841–845.

[174] A. L. Barabasi, “The origin of bursts and heavy tails in human dynamics”, *Nature*, vol. 435, no. 6, pp. 207–211, 2005.

[175] J. E. Hirsch, “An index to quantify an individual’s scientific research output”, *Proceedings of the National Academy of Sciences*, vol. 102, no. 46, pp. 16 569–16 572, 2005.

[176] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek, “Uncovering the overlapping community structure of complex networks in nature and society”, *Nature*, vol. 435, pp. 814–818, 2005.

[177] R. Baeza-Yates and E. Davis, “Web page ranking using link attributes”, in *Proceedings of the ACM International World Wide Web Conference (WWW)*, 2004, pp. 328–329.

[178] T. M. Chen and J.-M. Robert, “Worm epidemics in high-speed networks”, *IEEE Computer magazine*, vol. 37, no. 6, pp. 48–53, 2004.

[179] J. Dean and S. Ghemawat, “MapReduce: Simplified data processing on large clusters”, in *Proceedings of the USENIX/ACM Symposium on Operating Systems Design and Implementation (OSDI)*, 2004, pp. 137–150.

[180] S. A. Khayam and H. Radha, “Analyzing the spread of active worms over VANET”, in *Proceedings of the ACM International Workshop on Vehicular Ad hoc Networks (VANET)*, 2004, pp. 86–87.

[181] R. Cohen, S. Havlin, and D. ben-Avraham, “Efficient immunization strategies for computer networks and populations”, *Physical Review Letters*, vol. 91, p. 24, 2003.

[182] I. Gaber and Y. Mansour, “Centralized broadcast in multihop radio networks”, *Journal of Algorithms*, vol. 46, pp. 1–20, 2003.

[183] D. Kempe, J. Kleinberg, and E. Tardos, “Maximizing the spread of influence through a social network”, in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2003, pp. 137–146.

[184] P. Domingos and M. Richardson, “Proceedings of the seventh international conference on knowledge discovery and data mining”, in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2002, pp. 57–66.

[185] J. R. Douceur, “The sybil attack”, in *Proceedings of the International Workshop on Peer-to-Peer Systems (IPTPS)*, 2002, pp. 251–206.

[186] V. E. Krebs, “Uncloaking terrorist networks”, *First Monday*, vol. 7, no. 4, 2002.

[187] R. Fagin, A. Lotem, and M. Naor, “Optimal aggregation algorithms for middleware”, in *Proceedings of the ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*, 2001, pp. 102–113.

[188] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, “Optimized link state routing protocol for ad hoc networks”, in *IEEE International Multi topic Conference*, 2001, pp. 62–68.

[189] J. K. Kleinberg, “Authoritative sources in a hyperlinked environment”, *Journal of the ACM*, vol. 46, no. 5, pp. 604–632, 1999.

[190] S. Y. Ni, Y. C. Tseng, Y. S. Chen, and J. P. Sheu, “The broadcast storm problem in a mobile ad hoc network”, in *ACM/IEEE MOBICOM*, 1999, pp. 151–162.

[191] S. Brin and L. Page, “The anatomy of large scale hypertextual Web search engine”, *Computer Networks and ISDN Systems*, vol. 30, no. 1-7, pp. 107–117, 1998.

[192] R. M. Anderson and R. M. May, *Infectious Diseases of Humans: Dynamics and Control*. Oxford University Press, 1992.

[193] S. L. Feld, “Why your friends have more friends than you do”, *American Journal of Sociology*, vol. 96, no. 6, pp. 1464–1477, 1991.

[194] L. C. Freeman, “Centrality in social networks: Conceptual clarification”, *Social Networks*, vol. 1, pp. 215–239, 1978.

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

[196] M. G. Kendall, “A new measure of rank correlation”, *Biometrika*, vol. 80, pp. 81–93, 1938.