8-26-2022

Protection against Contagion in Complex Networks

Pegah Hozhabrierdi
Syracuse University, pegah.hozhabri@gmail.com

Abstract

In real-world complex networks, harmful spreads, commonly known as contagions, are common and can potentially lead to catastrophic events if uncontrolled. Some examples include pandemics, network attacks on crucial infrastructure systems, and the propagation of misinformation or radical ideas. Thus, it is critical to study the protective measures that inhibit or eliminate contagion in these networks. This is known as the network protection problem.

The network protection problem investigates the most efficient graph manipulations (e.g., node and/or edge removal or addition) to protect a certain set of nodes known as critical nodes. There are two types of critical nodes: (1) predefined, based on their importance to the functionality of the network; (2) unknown, whose importance depends on their location in the network structure. For both of these groups and with no assumption on the contagion dynamics, I address three major shortcomings in the current network protection research: namely, scalability, imprecise evaluation metric, and assumption on global graph knowledge.

First, to address the scalability issue, I show that local community information affects contagion paths through characteristic path length. The relationship between the two suggests that, instead of global network manipulations, we can disrupt the contagion paths by manipulating the local community of critical nodes.

Next, I study network protection of predefined critical nodes against targeted contagion attacks with access to partial network information only. I propose the *CoVerD* protection algorithm that is fast and successfully increases the attacker's effort for reaching the target nodes by 3 to 10 times compared to the next best-performing benchmark.

Finally, I study the more sophisticated problem of protecting unknown critical nodes in the context of biological contagions, with partial and no knowledge of network structure. In the presence of partial network information, I show that strategies based on immediate neighborhood information give the best trade-off between performance and cost. In the presence of no network information, I propose a dynamic algorithm, *ComMit*, that works within a limited budget and enforces bursts of short-term restriction on small communities instead of long-term isolation of unaffected individuals. In comparison to baselines, *ComMit* reduces the peak of infection by 73% and shortens the duration of infection by 90%, even for persistent spreads.

Protection against Contagion in Complex Networks

by

Pegah Hozhabrierdi
B.Sc., K. N. Toosi University of Technology, 2016
M.Sc., Syracuse University, 2020

Dissertation

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer & Information Science & Engineering.

Syracuse University
August 2022

To my best companion, the friend whom I always sought, and my strength in the darkest of days, my beloved Dunai.

Acknowledgements

Every journey is a an unpredictable blend of victories and defeats. What decides the final outcome is how one rises after a fall. I got to stand up tall after my share of defeats because of the support and generous helping hand of Dr. Sucheta Soundarajan, without whom I most probably have given up on this journey long ago. Her encouragements and trust taught me strength and autonomy over my research. Any young researcher would be lucky to receive this amount of support and understanding. I am indebted to her, not only for helping me to finish this journey with a bright smile, but also for the lessons I will carry with me for a long time to come.

I owe my first practical lessons in research and publishing to Dr. Chilukuri Mohan, with whom I had my first published work. The trajectory of my research and mindset changed drastically after working with him and his guidance opened new horizons for me. For lending me his experience and fatherly advise, I am forever grateful.

I have learned the most valuable currency in academia to be time, specially for young minds who are constantly thirsty for feedback from their seniors. To this end, I sincerely thank my defense committee members; Dr. Oh, Dr. Katz, Dr. Introne, Dr. Fioretto, and Dr. Mohan for taking time off of their busy schedule to go over my thesis and give me feedback. I especially thank Dr. Introne for his enthusiasm and valuable help during the different stages of my research, and members of SUNS lab for their team spirit and generous help whenever I needed a hand.

Stories cannot end without revealing the heroes in disguise. The heroes of my Ph.D. story are my family members who, from miles away, never lost their faith in me, no matter the circumstances. One hero, specifically, had an everlasting impact on my personal and professional growth and that is the person whom this work is dedicated to; my husband and friend, Dunai. His devotion, kindness, patience, constructive criticism, and valuable insight have changed me for the better and they continue to do so each passing day. I impatiently await to see what we both make of our journey together in the new chapters to come.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A social system, made of interacting components (also referred to as actors or agents), relies on certain (often simple) set of rules [2]. To facilitate mathematical analysis, these systems– whether in whole or in part– are modeled as networks. In networks, the actors are designated as nodes and the pairwise message passing relation between them as edges (also referred to as links or ties). For instance, human interactions in a group, the communication between computers in a network server, and the synaptic transmission between neurons are all built upon message passing rules between two adjacent components. The emergent collective behavior following such simple rules, however, are often complex[1] to model and predict [2]. The adoption of new ideologies, the evolution of swarm intelligence, and the rise and fall of epidemics in human social systems are some examples of complex collective behaviors that have been studied extensively for the past decades.

Spreading phenomena are key factors in the development of complex collective behavior and, as a result, the change in the dynamics and structure of a social system. A spreading process can either reinforce or resist an impact on the system. For example, a spread can characterize a viral infection or a prophylactic measure to prevent that infection. Depending on whether spreading processes reinforce a needed impact or propagate a harmful influence, they can be divided into *constructive* and *destructive* spreads. In literature, the destructive spreads are also referred to as *contagions*, a terminology that I will use in this study as well.

[1] I use the word *complex* in opposition to *trivial* and *expected*. For example, complex features are those contrary to trivial features of simple networks such as random, planar, and acyclic graphs.

Contagions can have catastrophic impacts on societies and infrastructure that people's lives and vitality depend on. For example, the outbreak of the COVID-19 pandemic affected billions of people worldwide and took the lives of millions.[2] Similarly, targeted attacks on critical infrastructure systems such as power grids and water treatments can jeopardize the operation of many crucial sectors, such as hospitals and food suppliers. For these reasons, it is important to study practical methods that can mitigate, reduce, or eliminate contagions.

Inhibiting a contagion in a network is known as the network protection problem. This problem investigates network modifications, in the form of adding or removing edges, that protect a set of critical nodes. These nodes are either known a priori based on their importance to the overall functionality of the network (predefined critical nodes), or have to be identified based on their contribution to the final size of the contagion (unknown critical nodes). The identification of the latter requires ranking methods that are based on the location of nodes in the network structure or the interaction among nodes in a group (community).

The focus of this work is on protection strategies that inhibit a contagion in real-world settings. More specifically, I study the network protection problem for both predefined and unknown critical nodes and offer algorithms that (1) are fast to compute and scalable; (2) perform within a limited budget; (3) use precise evaluation metric; and (4) do not rely on full network information.

1.1 Research Objectives

The main goal of this dissertation is to address some of the major shortcomings of the current network protection algorithms. Considering the importance of inhibiting contagions in real-world scenarios, my goal is to focus on practicality of protection algorithms in the face of a real emergency (e.g., propagation of unknown diseases and threats).

One of the main drawbacks of the existing network protection algorithms is their large computational cost, which leads to scalability issues. The reason is their reliance on global network measures that not only are computationally demanding, but also require the full network structure to be computed. I address this issue in in Chapter 3. More

[2] Source: https://covid19.who.int/.

specifically, I investigate the network properties and metrics that are known to impact the contagion flow. Through theoretical and empirical analysis, I prove the relationship between computationally inexpensive local network measures and the paths in the network through which the flow persists.

The local measures only depend on the immediate neighborhood of a node in the network. The result from Chapter 3 implies that for controlling the path of a contagion, we do not need to rely on global properties of the network. Although the network global properties contain more information about the contagion pathways, in practical settings, they are hard or impossible to compute (as I will show in the subsequent chapters). For example, in the face of a pandemic, the underlying network structure can only be known locally and through sampling from the population. My analysis in this chapter offers an answer to the challenging task of network protection against contagion in the presence of limited network information.

Another common issues in the study of network protection are using imprecise evaluation metrics and disregard for the implementation cost of the algorithm. I address this problem next by introducing a limited protection budget. I show that the combination of this budget and final contagion size gives a better evaluation metric than those used in the state-of-the-art.

More concretely, in Chapter 4, I introduce network modifications that protect a set of (predefined) target nodes (i.e., nodes whose security is of importance to the overall network) by restricting the contagion flow. This protection scenario is especially of interest for preserving privacy of some nodes in an infrastructure, such as World Wide Web or peer-to-peer networks. In this chapter, I propose the *CoVerD* algorithm, which relies only on local community (group) information of each target node, and aims to make a trade-off between the protection budget and magnitude of contagion. Despite relying on this limited information, *CoVerD* is shown to be more effective in protecting the targets than the state-of-the-art models, some of which relying on global properties of the network.

In Chapter 5, I address the problems of imprecise evaluation metric and reliance on global network measures for a more sophisticated setting, in which the critical nodes are not predefined. This type of protection is crucial in the face of biological contagions that can lead to epidemics and pandemics. Through a comparative analysis among different protection strategies, I show that the best trade-off between the efficacy and

cost of protection is achieved by relying on the immediate neighborhood information of each node. This reinforces the results obtained from Chapters 3 and 4.

Local information, however, are not always readily available. In fact, there are scenarios in which there is no prior information about the network structure available. Many of the studies on the network protection problem fail to consider this challenging scenario and assume full knowledge of the network structure.

In Chapter 6, I introduce the problem of protection against contagion in the presence of blindness towards network structure. In this setting, the protector does not have any information about the connections between nodes in the network. Hence, even using local information to influence the contagion is not possible. This is particularly a hard problem when no information on the dynamic of the contagion is known (e.g., protecting a large population against an unknown viral spread). To tackle this problem, I propose the *ComMit* algorithm. *ComMit* has two purpose: (1) obtaining information about the underlying network structure, and (2) introducing network modifications that inhibit the contagion flow. The dynamic nature of *ComMit* allows for partial information collection at each step and enhancement of previous modifications based on the newly acquired information. The blindness problem is tied with the problem of noisy data collection. In the same chapter, I also study the impact of noisy data on protection and the challenges it poses to *ComMit* algorithm.

1.2 Contributions

The major contributions of this work are as follows,

- I study the network measures that influence the paths through which contagion flows and show the relationship between local network measures and these paths, analytically and empirically.
- I formally define the privacy-preserving problem of protecting a set of nodes against an intruding contagion and propose the *CoVerD* algorithm as a scalable measure that surpasses the state-of-the-art methods.
- I introduce the problem of protecting the entirety of a network by inhibiting or eliminating a contagion in its early stages. The results of my analysis show that

the best trade-off between protection cost and efficacy is achieved through relying on local network information.

- Finally, I formulate and address the problem of network protection with unknown network structure. I propose the *ComMit* algorithm that dynamically and with a limited budget samples from the network and updates the network perturbations to inhibit the spread. I also touch on the problem of noisy data collection and robustness of *ComMit* in this scenario.

1.3 Dissertation Road Map

The first part of **Chapter 2** offers an overview of graphs and metrics used in graph theory for describing different network properties. In the second part, I examine the related work in spreading processes, contagions, and the network protection problem. In the last section, I explain the shortcomings of these studies (which are addressed in this research) and their implications in real-world applications.

Dependency on local network information (as opposed to global information) is the foundation of my proposed protection algorithms in this study. In **Chapter 3**, I prove theoretically and empirically that the path lengths in a small-world network are tied to the local clustering coefficient. The latter is a local property of the network and a measure of community-forming tendency among the nodes. The results of this chapter justify why the protection algorithms proposed in the subsequent chapters work; they use the local community information to find minimal network perturbations that influence the contagion paths.

In **Chapter 4**, I formally define the problem of vertex defense against crawling adversaries (network attack contagion) in a network, and show the shortcomings of the current protection strategies in the general setting. Next, I propose a defense algorithm, *CoVerD*, that only uses local community structure of a node and surpasses both global and local neighborhood perturbations.

Chapter 5 focuses on biological contagion of viral spread. The results of this chapter reinforce the fact that the local information on immediate neighborhood is sufficient and provides the best contagion inhibiting strategy with limited budget. Also, inspired by superiority of the community-based strategy in 4, I show that community-based approaches can also be successfully applied for designing effective mitigation strategies.

In **Chapter 6**, I introduce the problem of protection in a blind setting (i.e., no prior knowledge on network structure) and propose a dynamic mitigation that relies on information from geo-location network instead of the contact network, and uses an exploration-exploitation scheme to design an optimal testing strategy.

Chapter 2

Background

In this chapter, I lay out the theoretical foundation of my work and discuss the related work. First, I go through fundamental concepts in graph theory that are used in this research. More specifically, I introduce graphs as a mathematical structure, define the graph metrics required for understanding the remainder of this dissertation, and touch on the community structure in graphs and some of the related community finding algorithms. Second, I discuss the relevant literature for three categories of research on spreading processes: modeling, impact, detection, and protection/reinforcement. Third, I elaborate further on similar research categories addressing specifically the protection problem, which deals with contagion (i.e., destructive spreading phenomena). Finally, I discuss the major shortcomings of the current literature on the network protection, which built the motivation for and are addressed by the current study.

2.1 Graphs

Many real-world interaction-based systems can be conveniently described through a set of points and lines joining certain pairs among these points. This representation is a mathematical abstraction known as a graph. The friendship relationship between individuals, the communication between systems in a peer-to-peer network, and a power grid are some famous examples of systems that benefit from graph representation [3].

More formally, a graph G is an ordered pair $(V(G), E(G))$, in which $V(G)$ and $E(G) \subseteq V(G)^2$ are two disjoint sets and represent the set of points and the connection between points, respectively. When context is clear, we can just use (V, E). Commonly used

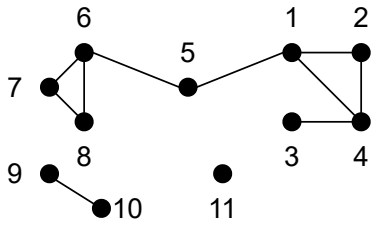


FIGURE 2.1: *The graph on* $V = \{1, ..., 11\}$ *with edge set* E ={(1, 2), (1, 4), (1, 5), (2, 4), (3, 4), (5, 6), (6, 7), (7, 8), (9, 10)}.

terms for points in V are *nodes* and *vertices*. The connections in E are also referred to as *edges* or *lines* [4]. A graph with no edges is an *empty* graph, whereas a graph with connection between all of its node pairs is a *complete* graph or a *clique*. An example of a graph is shown in Figure 2.1. Note that in this graph the nature of the connection between each pair of node does not matter and only the existence of a connection counts.

However, there are ways to include more information on the connections (edges). If for all $(v_i, v_j) = e_{ij} \in E$, we have $(v_i, v_j) = (v_j, v_i)$, the graph is considered *undirected*. Otherwise, it is a *directed* graph and the connection between v_i and v_j is shown with an arrow: $v_i \rightarrow v_j$. We can also consider the strength of a connection by attributing weights to each edge. In this case, we have a *weighted* or an *attributed* graph, referred to as $G = (V, E, W)$, in which $W = \{w_{e_{ij}} | e_{ij} \in E\}$ and $w_{e_{ij}}$ (also used as w_{ij}) is the weight of the connection between v_i and v_j. The graph in Figure 2.1 is an example of an unweighted, undirected graph.

Two vertices v_i and v_j are *adjacent* or *neighbors* if $(v_i, v_j) \in E$. We can summarize all adjacency relationships in the graph in a $|V| \times |V|$ matrix $\mathbb{A}$, called adjacency matrix, such that,

$$\mathbb{A}_{ij} = \begin{cases} 1, & (v_i, v_j) \in E \\ 0, & otherwise. \end{cases} \tag{2.1}$$

The number of neighbors of a node v_i (i.e., the sum of i^{th} row in $\mathbb{A}$) gives the *degree* of v_i and is referred to as $\mathbf{d}(v_i)$. A node of degree 0 is an *isolated* node (e.g., node 11 in 2.1). Note that a node can be adjacent to itself as well and form a *loop*. Since for spreading

phenomena, spreading to oneself or to no one is not meaningful or interesting, loops and isolated nodes are not considered in this study.

If we only consider a subset of nodes and/or edges of G, the resulting graph is a *subgraph* of G and is referred to as $G' \subseteq G$. In this case, G is the *supergraph* of G'. For example, $V' = \{1, 2, 3, 4\}$ and $E' = \{(1, 2), (3, 4)\}$ forms a subset of the graph in Figure 2.1.

A path of length k is a graph $P = (V(P), E(P))$ of the form $V(P) = \{v_1, ..., v_{k+1}\}$ and $E(P) = \{(v_1, v_2), (v_2, v_3), ..., (v_k, v_{k+1})\}$, in which P is non-empty and all v_i are distinct. The path of length 0 is a graph with one isolated node (*singleton* graph). In a graph $G = (V, E)$, there are one or more paths of varying lengths starting from each node. Informally, a path in G is a sequence of edges that connect two nodes. For example in Figure 2.1, $\{(1, 2, 4, 3), (1, 4, 3)\}$ are two paths connecting node 1 to 3.

As I will discuss in Chapter 3, when studying spreading phenomena, the *shortest paths* are the most informative. In fact, shortest paths are also referred to as *information pathways* [5] as it has been shown the flow of information travels through these paths. The length of the shortest path among all paths between two nodes is the *distance* between those nodes. For example, the shortest path between 1 and 3 in 2.1 is $\{(1, 4, 3)\}$, giving a distance of 2. The $k - hop$ neighborhood of a node v_i is the set of all nodes whose distance from v_i is k. For example, the $3 - hop$ neighborhood of 1 in 2.1 is $\{7, 8\}$.

$G = (V, E)$ is considered *connected* if it is non-empty and there is a path between any two vertices in V. A maximal connected subgraph of G is a *(connected) component* of G. The graph in 2.1 is a disconnected graph and has three connected components: $\{1, 2, 3, 4, 5, 6, 7, 8\}$, $\{9, 10\}$, and $\{11\}$. An edge whose removal increases the number of connected components in the graph is a *bridge* or *cut edge*. A vertex seperating two other vertices that belong to the same component is a *cut* or *bridge node*. In 2.1, 5 is a cut and both $(1, 5)$ and $(5, 6)$ are bridges. As will be discussed in the subsequent chapters, bridges and cuts act as super-spreaders in certain settings.

2.2 Community Structure

Nodes in a real network often show a tendency to group together. This feature of networks is known as *community structure* or *clustering* [6]. There are many definitions for a community depending on the context in which they are being studied. On a high-level,

a community structure is defined based on (1) the topology of the underlying graph (i.e., nodes in the same community are strongly connected), or (2) the attributes of the nodes in the graph (i.e., sharing a community means sharing a similar feature or role) [7]. Note that these two definitions are not exclusive and an attribute-based community can be a topological community as well. For example, in a network of elementary school friendship, in which nodes are students and edges are friendship among them, the attribute of class and friendship topology give similar communities as students in the same class have a higher tendency to befriend each other [8]. Since contagions (and in general, the spreading phenomena) are modeled as a node traversal process, their flow is heavily influenced by the network topology [9–13]. As such, whenever the notion of community structure is used in the current study, it is referring to topological community structure.

Communities can be overlapping, hierarchical, or non-overlapping. In most cases, however, they are defined as non-overlapping set of nodes that share more edges inside their community than outside [7]. This dissertation also considers this type of community.

The field of community detection is vast and deserves its own separate study. In general, these approaches are based on either of these three methods: modularity maximization [14–17], statistical inference [18, 19], and random walks [20, 21]. Among these, the most commonly used approaches are modularity maximization methods. Modularity, first proposed by Girvan and Newman [22] as a measure of community strength, calculates the difference between fraction of edges existing in a community and expected fraction if edges were distributed randomly. One of the most popular modularity-based community detection methods is Louvain algorithm [14], which, despite its shortcomings (namely instability of results and possibility of yielding disconnected communities) [23], remains one of the most effective and efficient methods available.

2.3 Graph Metrics

The three metrics used in analyzing the protection algorithms in this research are centrality measures, characteristic path length, and local clustering coefficient. Here, I briefly go through each.

Centrality Measures. In real world, a protection problem is often limited by a budget and faced with the critical decision of how to prioritize nodes for protection. The node importance ranking is usually obtained from a certain node centrality measure in the network. A centrality measure is a function on the set of nodes V whose output assigns

a number (ranking) to each node. Depending on how this function is defined, it is possible for several nodes to have the same ranking. As the definition of *importance* is subjective, different centrality measures have been used in the literature on spreading phenomena; namely *degree centrality*, *eigenvector centrality*, *betweenness centrality*, and *closeness centrality*.

Degree centrality for each node $v_i \in V$ is defined as its normalized degree with respect to the maximum possible degree in the network, i.e., $\frac{\mathbf{d}(v_i)}{|V|-1}$. This centrality assumes important nodes have more connections. Although this assumption has led to some powerful protection strategies [24] to combat the flow of a contagion, this is not always the case. For example, it has been shown that in networks with high modularity, the contagion flow is more efficiently suppressed through bridge nodes (which often have low degree centrality) than those with highest degree centrality [25]. The most powerful feature of degree centrality is that it is a *local* graph metric, meaning that it can be computed without global knowledge of the graph structure.

Eigenvector centrality attributes the importance of a node to the number of connections to other important nodes. For node v_i, it is defined as the i^{th} element of x such that $\mathbb{A}.x = \lambda.x$. Eigenvector centrality is shown to improve some degree centrality-based protection methods [26], however, its computation cost as a *global* measure is the biggest bottleneck for iterative protection algorithms.

Betweenness centrality, defined as the accumulative fraction of pairwise shortest paths in the network that pass through a node, considers the importance of bridge nodes as suggested in [25]. Despite its popularity in the network protection literature [27, 28], the reliance of betweenness centrality on computing all pairwise shortest paths in the network makes it an impractical tool for real-world problems.

Closeness centrality also relies on computing shortest paths. Instead of considering all the shortest paths that pass through the node v_i, it computes the reciprocal of the average shortest path length from v_i to all other nodes. Considering its similarity to betweenness centrality, it has also been popular in network protection studies [24, 27, 29], despite its expensive computation cost.

Characteristic Path Length. Similar to eigenvector, betweenness, and closeness centrality, characteristic path length is also a global graph measure. It is defined as the average of all pairwise shortest paths in the graph. Larger characteristic path length implies greater travel time for information flow [30]. The efficiency of a spread is also shown to be dependent on the characteristic path length [31].

Local Clustering Coefficient. First proposed by Watts and Strogatz [1], local clustering coefficient is the measure of a node's tendency to form a cluster. It considers the neighborhood of a node and how close they are to form a complete graph. The local clustering coefficient of a node v_i is defined as the ration between the number of existing connections among v_i's neighbors and all possible such connections. Evident from the name, this is a local graph measure which requires considerably less computational cost compared to global measures. It is mainly defined for undirected graphs, however, there are extensions to directed graphs as well [32].

2.4 Research Directions in Spreading Processes

A spreading phenomenon is defined as a process in which a node can influence its neighboring nodes to change their status. For contagion, this influence is destructive (e.g., diseases). Analysis of different spreading processes in a social network have been addressed from four main directions: modeling, impact, detection, and protection/reinforcement. Here, I discuss each of them in the context of relevant literature.

2.4.1 Modeling

The first scientific endeavour in understanding spreading processes started with modeling the dynamics of a spread. The mathematical models for spreading were first proposed by researchers in epidemiology [33] and sociology [34, 35], dating back to early twentieth century. Later studies have expanded these models to account for novel types of spread discovered in real world. The pivot of all these models is the spreading rule by which an "information" is passed from one individual to another. This "information" can be a disease, news, rumor, or anything that changes the state of the receiving end as a result. The spreading rules fall into two categories: independent cascade models, and threshold models.

Independent Cascade Models. The assumption behind these models is that each interaction between two agents (nodes) results in contagion with independent probability. More formally, if we consider an affected node v_i and an unaffected node v_j, the probability of v_i turning v_j into an affected node through their interaction is $p_{v_i v_j}$, or p_{ij} for short. Note that p_{ij} only depends on the interaction between v_i and v_j and is independent

from interaction with other nodes in the network. More frequent interaction between v_i and v_j increases p_{ij}.

Some of the well-known independent cascade models are susceptible-infected-recovered (SIR) and susceptible-infected-susceptible (SIS) from epidemiology; and Bass model from innovation diffusion literature. Here, I focus on SIR and SIS models, which are the most commonly used in the network protection studies. Interested reader may refer to [36] to find other proposed models in this category.

SIR and SIS model are emerged from epidemiology. Despite their simple spreading rules, they have been effective in modeling various spreading processes, such as diseases, idea propagation, and innovation diffusion. I explain their details in the context of an infectious disease. They both consider certain number of possible states for each node in the network at each point of time. For SIR model, these states are **S**usceptible, **I**nfectious, and **R**ecovered. For SIS model, there is only the two susceptible and infectious states. Susceptible individuals are healthy persons that may become infected through interaction with infected population. Infected people are those who are already carriers of the disease and can affect susceptible individuals.

In SIR model, the disease has a limited duration, after which the infected individual is considered recovered and immune to being infected again. The probability p_{ij} of an infectious node infecting a susceptible person is controlled by the infection rate α. The recovery of an infected node is controlled by a recovery probability β. In some variations, the recovery rate is considered deterministic and all infected individuals are recovered after a certain time. The SIS model does not assume the immunity after recovery and a recovered person becomes susceptible again. The controlling parameters of SIS is similar to SIR, except that β is $1 - \alpha$. Due to the immunity assumption, the infection will eventually die out in SIR model. On the other hand, in SIS model, the infection can reach an endemic state, where the fraction of infected population remains non-zero.

Threshold Models. In some scenarios, the probability of a node being affected depends on the frequency of interaction with more than one affected node. For example, in the spread of a social behavior, an individual is more likely to adapt the new behavior if a certain fraction of his/her connections have already done so [37, 38]. Intuitively, threshold models are independent cascade models with memory, in which an accumulation of the past and present interaction with affected nodes determines the probability of a node becoming affected.

The simplest threshold model is Linear Threshold Model (LTM). LTM is an information diffusion model in which the activation of a node v depends on its threshold θ_v and the sum of incoming edge weights from all activated neighbors of v. The node v gets activated if and only if this sum surpasses θ_v. Other notable threshold models are the voter model [39], the standard rumor model [40], and the strategic game model [41].

2.4.2 Impact

This field of study aims at finding the ways network topology impacts the dynamics of a spread. The majority of work on modeling spreading phenomena lies in this category

Ganesh et al. [9] investigate the impact of network topology on the spread of epidemics. Their goal is to, theoretically, identity the topological properties of a graph that determine the persistent of epidemics. This study is limited to certain synthetic graph structures (hypercubes, complete graphs, random graphs, stars, and power-law graphs) and does not involve real-world data. On these graphs, they find conditions that can stop a contagion either quickly (logarithmically in the size of the network) or slowly (exponentially in the size of the network).

Other works, which are centered around real-world networks, broach this problem through finding nodes that contribute the most to the propagation of a spread (also known as *super-spreaders*). These studies rely on various graph centrality measures to detect these nodes [42–44]. However, Pei et al. [45] raise the concern that relying on simulation-based spreading processes does not correctly identify such super-spreaders. Instead, they rely on real information diffusion data in online blogging social networks to infer the best centrality measure that correctly captures the super-spreaders. For these datasets, they find the k-core based centrality, which is a global measure, to have the best predictability. Most notably, they also propose the sum of the nearest neighbors' degree as a local proxy for measuring users' influence instead of the global k-core centrality. Serafino et al. [46] reach the same conclusion about k-core centrality by studying the GPS mobility data during the COVID-19 outbreak in Latin America.

2.4.3 Detection

The presence of a spread is not always as straightforward to detect as in the case of, for example, viral diseases. There are a number of controversies when it comes to

determining the existence of spread based on real-world data. For instance in economic studies, Lando et al. argue that the detection of contagion is only the result of poor statistical techniques [47]. This claim, however, has been refuted by other studies. For example, Favero et al. [48] comprehensively study the money markets of ERM member countries and fail to reject the null hypothesis of no contagion.

On the other hand, there is the challenge of noisy data. In the case of social networks, for example, the majority of detection techniques assume zero information loss in the data [49]. A number of recent studies have tackled the more realistic problem of detecting a spread in the presence of noisy data (e.g., incomplete, misleading, or misattributed data).

Meirom et al. [50] propose an epidemic detection algorithm that only uses the local neighborhood information. They show the success of their algorithm in different settings, such as in the presence of high false positive and false negative rates, partial network information, and more than a single "patient zero".

Milling et al. [51] extend the algorithm proposed by Meirom et al. to weighted networks. A weighted contact network implies the infection does not simply travel at the same speed between all connected nodes. They achieve a superior performance under the same settings (i.e., noisy data, partial network, and multiple infectious seeds).

2.4.4 Protection/Reinforcement

This field of research distinguishes itself from the previous two by building on the presupposition that (1) indeed, there exists one or multiple spreading phenomena in the network; and (2) there is a general awareness of the vulnerabilities within the network. The critical nodes in a network are either predefined or identified as those whose resistance threshold against a spread is low and contribute the most to the propagation of the spread.[1]

The main objective of the **protection problem** is to find the most efficient graph manipulation, as either introducing perturbations (i.e., removal or addition of edges and/or nodes) or changing node status (e.g., vaccination), to protect the critical nodes, by either blocking the spreading process or, in the case of persistent spread, minimizing the probability of spread reaching these nodes. The critical nodes are characterized either as

[1] critical nodes are also referred to as *susceptible* nodes in [52] and [53]. These terms will be used interchangeably throughout this study.

nodes of certain value (e.g., having high centrality [27, 45], containing sensitive information [54], or acting as super-spreaders [24, 27]), or as a set of nodes whose collective interaction marks their level of vulnerability (i.e., communities) [55, 56].

The dual problem to protection of critical nodes against a spread is the **reinforcement problem**, in which the objective is to find the *boosting nodes* that maximize the coverage of a spread (in contrast to the protection objective of minimizing this coverage). Some examples of reinforcement problem include viral marketing [57], innovation diffusion [58], engagement maximization [59], and reach maximization [60]. The difference between the two problems lies in their type of spread; while the **protection** scenario considers *contagion*, the **reinforcement** scheme targets an underlying *constructive spread.*

Considering the duality between *protection* and *reinforcement* problem, the results of this research can potentially be used in designing reinforcement strategies that maximize a constructive spread in the network. The importance of protection scenarios are best demonstrated through previous studies in: protection against targeted attacks on infrastructure [24, 29, 54], epidemic control and immunization [25, 61, 62], and prevention of misinformation propagation in online social networks (OSNs) [63].

Heuristic Design. The general reinforcement problem is equivalent to influence maximization. Kempe et al. [64] have shown that influence maximization in both Independent Cascade Model and Linear Threshold Model are NP-complete. Similarly, in various adaptations of the general protection problem, it has been shown that the protection problem entails the minimization of a non-monotone non-submodular function and is NP-complete [27, 54]. This suggests that the best achievable solution can be only in the form of a heuristic.

2.5 General Classification of Contagions in the Network Protection Problem

A contagion process can be simple or complex. In **simple contagion**, which is modeled by independent cascade models, the activation (or infection) of a single node is sufficient for the transmission via message passing. By contrast, in **complex contagion**, modeled by threshold models, a successful transmission requires contact with more than one activated node [65]. The contagion processes for the network protection problem can

be categorized into three general groups based on the context in which the spread starts: network attack contagions, biological contagions, and social contagions.

2.5.1 Network Attack Contagions

This group of spreading processes involve contagions that are artificially designed to attack a network, or perform unauthorized data collection, for a given budget. These are often *simple contagions* that follow a simple, yet efficient, algorithm to select the targets of the attack within the network.

Two notable examples of this contagion type are (1) crawling-based attacks on social and peer-to-peer networks for the purpose of user de-anonymization and/or unwarranted information access [24, 54]; (2) the cascading failures in power grids due to intentional or accidental[2] disruption in the transmission system [66].

The main challenge in protecting node(s) against such attacks is the lack of knowledge on the logistics of the attacker (i.e., the starting point of the attack, the crawling algorithm used, and the available attacking budget). Moreover, for scaleable and efficient defense against these attacks, the defense heuristic has to rely only on local neighborhood information (in contrast to global network measures, such as shortest path length or closeness centrality) [24, 29].

2.5.2 Biological Contagions

Possibly the most widely studied type of spreading phenomena, biological contagions include the epidemic models used for predicting and mitigating the outcome of viral spreads. Propagation of pathology through brain networks also fits in this category [67].

Biological contagions are often modeled as *simple contagions* [65], however, the models based on *complex contagion* are shown to have better generalizability [68]. Protection against these spreads are particularly challenging due to the dynamic nature of diseases [69], the co-infections or the coexistence of multiple contagions (of various types) in the network (i.e., the so called *ecology of complex contagions* [65]), and the lack of global knowledge of the underlying social (contact) network [25].

[2]The inadvertent cascading failure can be considered as an attack with a naïve attacking strategy, such as random node selection.

2.5.3 Social Contagions

Popularized through studies in sociology and viral marketing, social contagions have remained among the most complex processes to model and analyze. The propagation of beliefs [70], emotions [71, 72], rumors [73], opinions [74], misinformation [75], ideology [76], and behavior [77] shape the wide spectrum of the social contagion context. The multifaceted nature of social contagions, and the different contagion dynamics from one context to the next, amplifies the complexity of a general social contagion model.

To tackle the seemingly unattainable problem of modeling general social contagions, the studies focus only on one type of social contagion (e.g., rumor propagation) and use empirical characterization of human social interactions (e.g., the strength of weak ties [78], the small-world property [79], and assortativity [80]) to model the spread [68].

Social contagions pose another, and more unique, challenge in finding an optimal protection heuristic: the lack of control over individual's choice of forming social ties. More specifically, if an optimal perturbation scheme deduces the existence of an edge as harmful, there are no clear answers to *how* one can go about severing that connection between two individuals. As such, the majority of current research focus on either the analysis of the contagion process or the detection of the susceptible hubs.

In summary, contagions can be categorized based on either the **context** (i.e., attack on network infrastructure, biological processes, or social phenomena), or the **mechanism** of the spread (i.e., simple vs. complex contagion). This study covers network attacks and biological contagions by, first, outlining the challenges and major drawbacks in the state-of-the-art protection schemes, and, second, offering optimal heuristics that address those existing shortcomings.

2.6 The Shortcomings of Current Trends in Network Protection Research

From a general perspective, the majority, if not all, research addressing the three context-based contagion types suffer from one or more of the following shortcomings.

Assumption of global graph knowledge. In the case of real-world biological and social contagion scenarios, the knowledge of the underlying social (contact) network is

non-existent or severely limited. Combating a viral infection, the real-world problem setting of biological contagion, relies on testing strategies (e.g., contact tracing) that give noisy information about sporadic local neighborhood in the network, and this is the best estimate of the network structure that protector can have [81, 82]. A similar problem arises in the study of social contagions. In the case of face-to-face social networks, this problem is more pronounced as the existence of any sort of connection between two individuals is not known a priori and theoretical models are used to assess their presence. For example, Starnini et al. propose an agent-based modeling that reproduces quantitatively some features of face-to-face interaction networks through random walks [83]. However, even in online social networks in which the existence of different type of edges is known (e.g., follower-followee, retweet, and mention relationship in Twitter, friendships in Facebook, reply and Karma in Reddit, etc.), we are not aware of the importance of these connections.

This is the problem of defining *socially relevant connections* [84]; we might know, for example, whom a person A follows in online social media, or gets in touch with through Email or phone, but we cannot easily infer which of these connections are meaningful enough to leave an impression on A's social behavior and beliefs. Moreover, relying on global network structure begets the scalability bottleneck of network protection heuristics. For example, in the case of network attack contagions, the protection algorithms that rely on global centrality measures lose their efficacy rapidly as the size of the network (number of nodes and/or edges) increases [24, 27, 29].

Ignoring the dynamic nature of spread and/or network. Both the contagion process and the underlying network can be dynamic with time-dependent parameters (characteristics). For example, the infection rate of a viral spread is often time dependent [85], and so are the interaction between individuals (e.g., they can depend on days of the week and seasons, or be impacted by other contagion processes such as ideological shifts [86]). As the co-evolution of multiple dynamic processes with unknown covariates makes the protection problem intractable, many studies consider either the spread process or the network to be static. As long as the time scale of change in one of these two is slower than that of the other, the static assumption is adequate for real-world applications [87]. However, when these scales become comparable, it is necessary to consider the dynamic nature of both.

Imprecise evaluation metrics. More applicable to network attack contagions, there is no consensus on the definition of "successful protection" among different studies. The majority consider a certain centrality measure and propose methods for minimizing it.

Closeness centrality, betweenness centrality, degree centrality, and eigenvector centrality are the most popular choices [24, 27, 29, 88]. However, I show in Chapter 4 that minimizing these (often global) centralities do not necessarily decrease the probability of discovery by an attacker. Indeed, by defining the evaluation metric as the budget spent by an attacker, we can both reduce these centralities and the probability of discovery by the attacker.

Scalability. Contagion processes propagate through contagion paths in the network (see Chapter 3). The main goal of a protection algorithm is to introduce perturbations to the network such that these paths are either cut or elongated. Previous studies have shown the impact of global measures, in particular shortest path length, in determining the efficiency of a spread [31, 54]. Network centrality measures used in detecting the "super-spreaders" (in the form of bridge nodes) also depend on the shortest paths in the network (e.g., closeness centrality, harmonic centrality, and betweenness centrality). Hence, for influencing the contagion paths, the studies often focus on shortest paths and related global measures in the network. However, this severely limits the applicability of such methods in real-world scenarios as they do not scale well. The main question to be answered is whether we can influence the contagion paths as a global property of a network by focusing only on local information that are easy to acquire.

Chapter 3

Contagion Paths & Local Community Information

In real-world settings, mechanisms and dynamics of a contagion is often unknown. Similarly, the information on the underlying network structure may only be partially known. The current study is built on these two assumptions (see Chapters 4 and 6). To devise a protection scheme that goes beyond the specification of the underlying contagion model, we need to focus on network structural properties that impact the general behavior of a spreading phenomenon. Recall that a spreading phenomenon is defined as a process in which a node can influence its neighboring nodes to change their status. This local behavior translates into the global phenomenon of contagion in the network in which information (e.g., disease, news, beliefs, etc.) propagates through certain paths that I will refer to as *Contagion Paths*.

Large real-world social networks (including those discussed in this study) often exhibit small-world properties, i.e., a low average shortest path length (or characteristic path length, as defined in Section 3.2.2) and a high clustering coefficient [1]. The efficiency of contagion paths (i.e., magnitude of contagion) vary based on the underlying network structure; e.g., random network, scale-free network, and small-world network [31, 89, 90]. In particular, the small-world properties – characteristic path length and clusterability of the network structure – are shown to be paramount in determining the efficiency of the spread [31].

The characteristic path length is a global graph measure that requires full knowledge on the network structure and computing it has a time complexity of $O(mn)$, in which m and n represent the number of edges and nodes respectively. The clusterability, on the

other hand, can be expressed locally as local clustering coefficient (defined in 3.2.1) and can be computed in $O(n\sqrt{n})$ for all nodes (or $O(\sqrt{n})$ for each node). Having scalability at the heart of this study, my goal is to limit the proposed protection algorithms to local information. In this section, I will show, theoretically and empirically, that the global property of characteristic path length can be expressed in terms of local clustering coefficient for small-world networks. Hence, the efficiency of the contagion, which has been shown to depend on characteristic path length and clusterability, can be controlled by perturbations that target the local clustering coefficient of the network.

The importance of clustering coefficient, as the measure of nodes' tendency to group together, signifies the importance of community information in managing the magnitude of the spread. The result of this chapter on the importance of local community information, illustrated by the local clustering coefficient, pave the path for subsequent chapters in which I propose scalable and efficient protection algorithms against different types of contagion.

3.1 Related Work

Pinto et al. [90] studied the impact of network topology on the viral spread for several complex network models, namely random, small-world, scale-free, modular, and hierarchical network models. They model the viral spread as SIR model with and without vaccination. Their results show that the propagation of disease is heavily impacted by the underlying network structure. More specifically, the spread of disease slows down as the network becomes more modular (i.e., increasing clusterability). I will refer to studies that show the impact of modularity on the spread for a broader class of viral spread models in Chapter 6.

In a similar study, Shirley and Rushton [89] show the importance of shorter characteristic path length on increasing the magnitude of infection. They also emphasize the impact of local and global heterogeneity of network topology on reducing or increasing the efficiency of spread. More specifically, the local heterogeneity, represented by the presence of clusters, reduces the magnitude of spread only in the intermediate levels of clustering (i.e., where vertices clustering coefficient follows a non-uniform distribution). The global heterogeneity, characterized by the different contact pattern across all vertices, can increase the magnitude of infection if close to uniform distribution and

decrease it otherwise. The small-world networks introduce both local and global heterogeneity; the local heterogeneity through introducing non-uniform clustering while retaining a short characteristic path length, and global heterogeneity through random rewiring that leads to non-uniform local contacts.

Cowan and Jonard [31] go further and show the trade-off between the impact of path lengths and clustering coefficient (referred to as "cliquishness" in the paper) on the magnitude of contagion in the context of knowledge diffusion. Although the shorter path length are associated with higher contagion size, shortening these paths more than a certain level will destroy the clusters (and heterogeneity), which in turn reduces the contagion size. In summary, the addition of links (reducing path lengths) can only help the contagion size if the clustering structure is preserved. A region with sufficiently short path length and high clustering structure is the small-world region and close to the topology of real-world networks.

The studies that connect the network structure to spreading dynamics, as mentioned above, are context-based (e.g., network attack contagions, viral spread, rumor propagation, etc.) and not comprehensive in terms of the network measures considered. However, a common conclusion in all these studies is the importance of characteristic path length and clustering coefficient in controlling the size of a contagion in small worlds, regardless of contagion context. I take these studies one step further and consider the relationship between characteristic path length, as a global measure, and local network measures (such as clustering coefficient). The goal is to find a local estimator of this global measure that can help with realizing scalable protection strategies whose aim is to reduce the contagion size.

I first start by a theoretical analysis of the well-known Watts-Strogatz small-world model [1] in Section 3.2. Next, I will expand my analysis through a comprehensive empirical study on 10 real small-world networks in Section 3.3. The result of both analysis is summarized in 3.4 and paves the path for the next chapters.

3.2 Path Length & Clustering Coefficient in Small Worlds: Theoretical Analysis

The small-world experiment, conducted by Milgram's team in 1967 [79] (see Section 3.3 for more details), empirically demonstrated that the individuals in human societies,

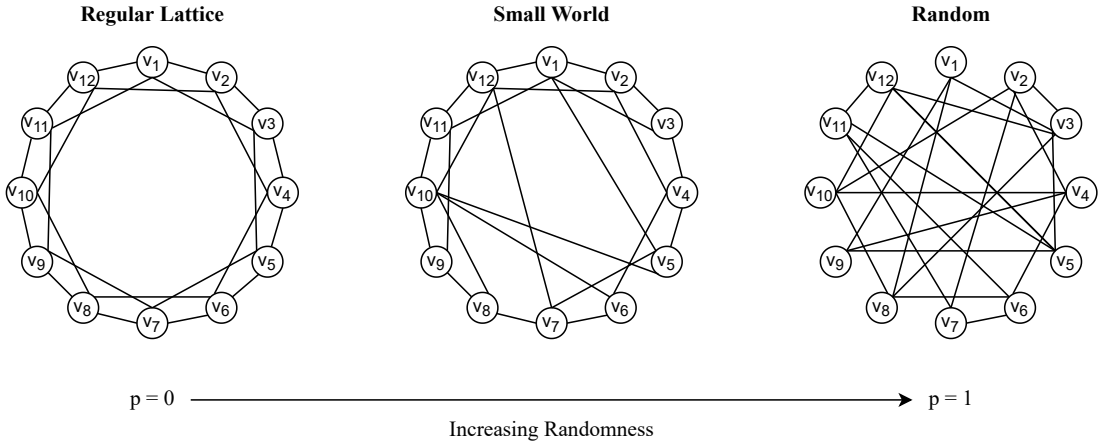


FIGURE 3.1: *Watts-Strogatz small-world model is an intermediary between regular lattice with large clustering coefficient and characteristic path length, and random graph with small clustering coefficient and characteristic path length. The amount of randomness introduced to the network is controlled by parameter p. Figure recreated from [1].*

on average, are connected via short paths of length six (six degrees of separation). In 1998, Watts and Strogatz introduced their small-world model that can successfully create the features of a small-world network; short characteristic path length and high clustering coefficient. Watts-Strogatz model starts from a regular lattice and introduces randomness to the lattice through rewiring mechanism controlled by parameter p.

A regular graph is one in which all nodes have the same degree. A regular lattice, as a special case of a regular graph, imposes an ordering on the nodes and limits the connections of a node to its nearest neighbors based on this ordering. To build a regular lattice of degree K (where K is even) on an ordered set of nodes $\{v_1, v_2, ..., v_n\}$, each node v_i is connected to $\frac{K}{2}$ nodes before and after itself in the ordered set. In other words, the edge (v_i, v_j) exists iff

$$0 < |i - j| \leq \frac{K}{2}. \tag{3.1}$$

Figure 3.1 shows a regular lattice of degree 4.

The Watts-Strogatz algorithm rewires each existing edge in the regular lattice with probability p. The higher this probability, the closer is the final network to a random graph, as shown in Figure 3.1. Since small-world model is the intermediary between regular

lattice and random network, its network measures are also bounded by their corresponding measure in the the lattice and random graph. Formally, if we use the subscript p for small-world model, 0 for regular lattice, and 1 for random graph (since $p = 0$ and $p = 1$ yield regular lattice and random graph, respectively), we have

$$\mathcal{L}_0 \leq \mathcal{L}_p \leq \mathcal{L}_1, \tag{3.2}$$

$$C_0 \leq C_p \leq C_1, \tag{3.3}$$

in which $\mathcal{L}$ and C represent characteristic path length (i.e., average shortest path length) and clustering coefficient (i.e., average local clustering coefficient) of the network, accordingly.

In the following, I prove the relationship between the global measure of characteristic path length and local clustering coefficient in Watts-Strogatz model. To do so, I leverage the relationship between structural properties of small-world and regular lattice; i.e., $\frac{\mathcal{L}_p}{\mathcal{L}_0}$ and $\frac{C_p}{C_0}$.

3.2.1 Clustering Coefficient

Consider an undirected[1] graph $G = (V, E)$ with $V = \{v_1, ..., v_n\}$ and $E = \{(v_i, v_j)|v_i, v_j \in V\}$ representing the set of vertices and edges respectively. Recall that the local clustering coefficient of a node $v_i \in V$ is a measure of connectivity between its neighbors and is defined as the fraction of existing connection among v_i's neighbors and all possible such connections. Formally, if we define the neighborhood graph of v_i as $G_i = (V_i, E_i)$ in which $V_i = \{v_j|(v_i, v_j) \in E\}$ and $E_i = \{(v_j, v_k)|v_j, v_k \in V_i, (v_j, v_k) \in E\}$. The number of possible connections between neighbors of v_i (i.e., nodes in V_i) is $\binom{|V_i|}{2}$. Hence, the local clustering coefficient of v_i is $\frac{|E_i|}{\binom{|V_i|}{2}}$. The clustering coefficient for G is defined as the mean of all local clustering coefficients for its vertices; i.e.,

$$C(G) = \frac{1}{|V|} \sum_{i=1}^{n} \frac{|E_i|}{\binom{|V_i|}{2}}. \tag{3.4}$$

[1] In this chapter I focus on undirected networks as justified in Section 3.3. However, all definitions hold for directed graphs as well with minor adjustments.

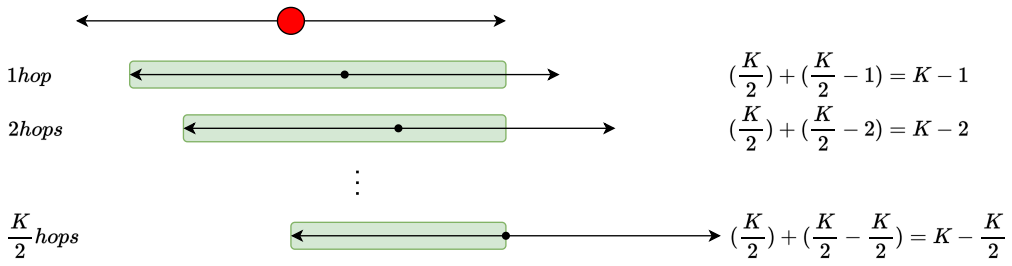


FIGURE 3.2: *The number of edges within the neighborhood of node v_i (shown in red) for h hop away from v_i is $K - h$ for $h \leq \frac{K}{2}$. The first row shows v_i in red and the span of its neighborhood that includes $\frac{K}{2}$ nodes to the left and right (total length is K). The figure shows h hops away within the $\frac{K}{2}$ neighborhood to the right of v_i. The green area shows the number of edges in each hop away that reside in the neighborhood of v_i and the values on the right depict the length of each green area.*

This is equivalent to counting the number of triangles formed by $V_i \cup v_i$ and compare it to all possible triangles that could be formed by the same set of nodes.

Theorem 3.1. *The clustering coefficient of regular lattice of degree K is $\frac{3(K-2)}{4(K-1)}$.*

Proof. To understand this, recall that in a regular lattice each vertex v_i is connected to $\frac{K}{2}$ nodes before and after itself. If we limit ourselves to the $\frac{K}{2}$ neighborhood to the right of v_i (Figure 3.2), we see that at h hops away from v_i only $K - h$ edges (i.e., $K - h$ fraction of the degree) reside in the neighborhood of v_i. The same pattern hold for the $\frac{K}{2}$ neighborhood to the left of v_i. If we consider both left and right of v_i, we are counting each edge twice. So, it suffices to only count the number of edges in the right side of v_i's neighborhood that overlaps E_i. Substituting the corresponding values to 3.4, we have,

$$
\begin{aligned}
C_0 = \frac{1}{|V|} \sum_{i=1}^{|V|} \frac{(K-1) + (K-2) + ... + (K - \frac{K}{2})}{\binom{K}{2}} = \\
\frac{1}{|V|} \times |V| \times \frac{(\frac{K}{2} \times K) - (1 + 2 + ... + \frac{K}{2})}{\frac{K(K-1)}{2}} = \\
\frac{(\frac{K}{2} \times K) - (\frac{K}{4}(\frac{K}{2} + 1))}{\frac{K(K-1)}{2}} = \frac{3(K-2)}{4(K-1)}
\end{aligned}
\tag{3.5}
$$

□

There is an analytical relationship between C_p and C_0. Since this relationship is crucial to the final theoretical result, I prove this relationship here.

Theorem 3.2. *The clustering coefficient of a regular lattice of degree K and that of Watts-Strogatz small-world network resulting from such lattice are related as,*

$$C_p \approx (1-p)^3 C_0. \tag{3.6}$$

Proof. A triangle formed by v_i and two of its neighbors in the small-world network implies that these three edges have survived the rewiring process. There are two possible scenarios in which this can happen:

1. All three edges have not been rewired. This can happen with probability $(1-p)^3$.
2. Only $(3-k)$ edges have not been rewired and k edges are the result of rewiring. The probability of this happening is $p^k(1-p)^{3-k}$.

Thus,

$$C_p = C_0((1-p)^3 + \sum_{k=1}^{3} p^k(1-p)^{3-k}). \tag{3.7}$$

Since p is small for small-world networks (see Figure 3.3), the summation term is negligible in comparison and 3.6 is obtained. □

Figure 3.3 shows the accuracy of this approximation compared with values obtained from multiple simulations.

3.2.2 Characteristic Path Length

The characteristic path length is the average of all pairwise shortest paths in graph $G = (V, E)$ and is referred to as $\mathcal{L}(G)$. Formally,

$$\mathcal{L}(G) = \frac{1}{|V|(|V|-1)} \sum_{i=1}^{|V|} \sum_{j=i+1}^{|V|} \min \text{dist}(v_i, v_j). \tag{3.8}$$

Theorem 3.3. *The characteristic path length for a regular lattice of degree $K \geq 2$ with n nodes, where $n \gg K$, can be approximated as*

$$\mathcal{L}_0 \approx \frac{n}{2K}. \tag{3.9}$$

Proof. Since a regular lattice is symmetric, all nodes have the same average shortest path length to other nodes. Hence, the characteristic path length of the full lattice is equal to the average shortest path length from one node to rest. Let's consider that node to be v_i. Based on the definition of regular lattice in 3.1, h-hop away from v_i includes $\frac{K}{2}$ nodes; i.e., there are $\frac{K}{2}$ nodes whose shortest path length from v_i is h. The largest possible shortest path length in a regular lattice is $\lceil \frac{n}{K} \rceil$, so $h \leq \lceil \frac{n}{K} \rceil$.

The sum of all shortest paths from v_i to all nods in $\{v_j | i < j \leq \lceil \frac{n}{K} \rceil\}$ is $\frac{K}{2}(1+2+...+\lceil \frac{n}{K} \rceil)$. Using the symmetry of the lattice, the average shortest path length from v_i to $n-1$ other nodes in the lattice (i.e., the characteristic path length of lattice) becomes

$$\mathcal{L}_0 = 2 \times \frac{\frac{K}{2}(1+2+...+\lceil \frac{n}{K} \rceil)}{n-1} = \frac{K}{2(n-1)} \lceil \frac{n}{K} \rceil (\lceil \frac{n}{K} \rceil + 1) = \begin{cases} \frac{n(n+K)}{2K(n-1)}, & \text{even } n, \\ \frac{(n+1)(n+K+1)}{2K(n-1)}, & \text{odd } n. \end{cases} \tag{3.10}$$

For sufficiently large n, the result is approximately $\frac{n}{2K}$. □

A useful conclusion from Theorems 3.1 and 3.3 that I will use in next section is that

$$\frac{\mathcal{L}_0}{C_0} \approx \frac{\frac{n}{2K}}{\frac{3(K-2)}{4(K-1)}} \approx \frac{2n(K-1)}{3K(K-2)} \approx \frac{2n}{3K}. \tag{3.11}$$

Although there is no analytical relationship between $\mathcal{L}_p$ and $\mathcal{L}_0$, we can estimate $\frac{\mathcal{L}_p}{\mathcal{L}_0}$ for different levels of small-worldness for different values of p. Figure 3.3 shows the values for $\frac{\mathcal{L}_p}{\mathcal{L}_0}$ and $\frac{C_p}{C_0}$ for different p. The points are obtained by averaging through six networks of size 500, 1000, 1500, 2000, 2500, and 3000 nodes and degree 4, 10, 14, 20, 24, and 30 respectively. For $0.01 < p \leq 0.1$, the resulting network has the high clustering of a lattice with small path length of a random graph. This region is often referred to as *small-world region*. For $0 < p \leq 0.01$, the network has more lattice-like behavior and is referred to as *lattice-like region*. Similarly, we have a *random region* for $0.1 < p \leq 1$ in which the network has more similarity to a random graph with its low clustering and small path length.

Considering each of these three regions separately, we can find an approximation of $\frac{\mathcal{L}_p}{\mathcal{L}_0}$ in terms of p by finding the best fitted curve. I use non-linear least squares method

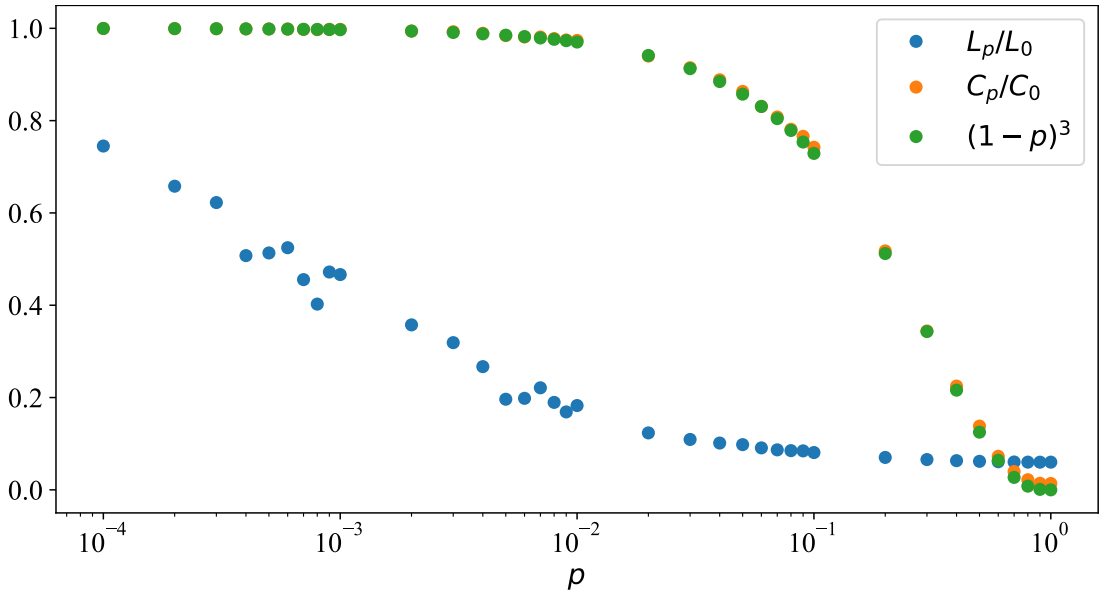


FIGURE 3.3: *The relative clustering coefficient, its approximation, and relative characteristic path length of Watts-Strogatz model for different values of p. The data points are obtained by averaging through six different networks with different sizes and degrees. The small-world region lies in* $0.01 < p \leq 0.1$ *in which the clustering is still large but shortest path is sufficiently small.*

to fitted curve. Figure 3.4 shows the best fit for each region. The quality of the fit is assessed by residual sum of squares (RSS) which is defined as

$$\text{RSS} = \sum_{i}^{n} (f(x_i) - y_i)^2, \tag{3.12}$$

and is reported for each fitted curve in Figure 3.4. The best fitted curves give the following relative characteristic path length for each region,

$$\frac{\mathcal{L}_p}{\mathcal{L}_0} \approx f(p) = \begin{cases} -0.12(3.5 + \ln(p)), & 0 < p \leq 0.01 \text{ (lattice-like)} \\ 0.07(1 - 8.22p)^3 + 0.08, & 0.01 < p \leq 0.1 \text{ (small-world)} \\ 0.03(2 + e^{-5.6p}), & 0.1 < p \leq 1 \text{ (random).} \end{cases} \tag{3.13}$$

This result will be used in next section to infer the relationship between the characteristic path length (as a global measure) and clustering coefficient (as a local measure).

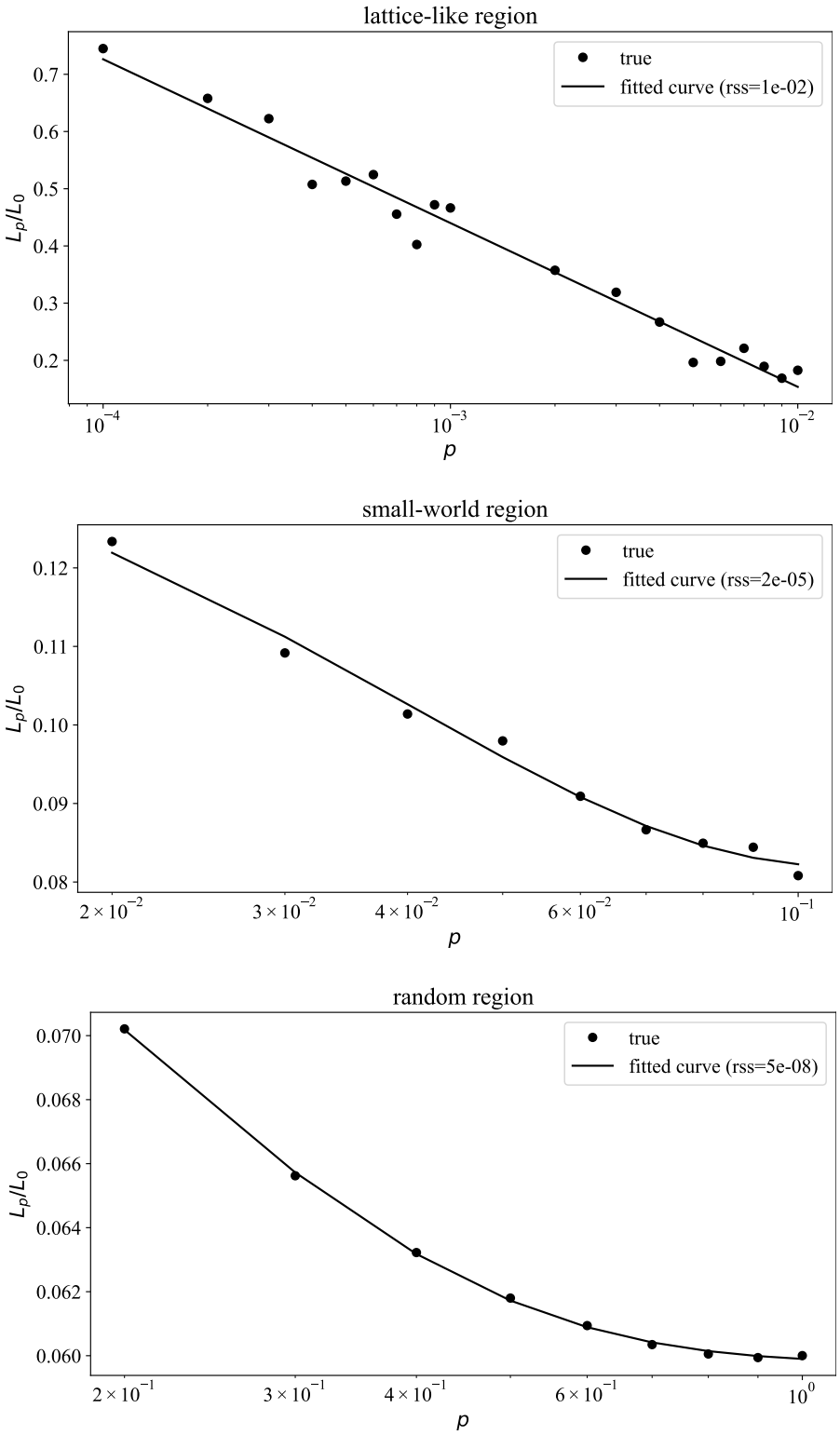


FIGURE 3.4: *The best fitted curve for relative characteristic path length in three different regions: lattice-like, small-world, and random. The exact function for each curve can be found in 3.13.*

3.2.3 Path Length Described by Clustering Coefficient

Here I combine the results in the previous two sections to establish the relationship between $\mathcal{L}_p$ and C_p as follows,

$$\frac{\mathcal{L}_p}{C_p} \overset{3.2}{\approx} \frac{\mathcal{L}_p}{(1-p)^3 C_0} \overset{3.11}{\approx} \frac{2n}{3K(1-p)^3} \times \frac{\mathcal{L}_p}{\mathcal{L}_0} \overset{3.13}{\approx} \frac{2n}{3K(1-p)^3} \times f(p) \tag{3.14}$$

Hence, there is an approximate relationship between clustering coefficient and characteristic path length in small-world networks that depends only on p. For small-world networks, this relationship is a polynomial fraction of same degree. This shows that, theoretically, the contagion paths in small worlds, which are shown to correlate with small-world properties, can be influenced by focusing on the local clustering (i.e., structural community) information alone instead of incorporating global measures, such as shortest path lengths.

The biggest drawback of the Watts-Strogatz model is its inability to imitate the scale-free degree distribution of real-world networks. In the remaining of this chapter, I set up an empirical study to show that in real-world scenarios there also exists a relationship between the clustering coefficient and characteristic path length. Although this relationship is not as straightforward as in 3.14, it demonstrates the possibility of influencing shortest paths via local clustering information nonetheless.

3.3 Path Length & Clustering Coefficient in Small Worlds: Empirical Analysis

Network connectivity and shortest paths have been widely used to study information diffusion and rumor propagation [91]. Shortest paths provide the fastest and, usually, the strongest interaction between actors (nodes) in a network [92]. Theoretical studies on shortest paths in social networks often neglect one of the most well-known properties of these networks: small-world phenomenon [93–95]. Small-world phenomenon, first popularized by Milgram in the 60's [79], indicates that individuals in a social network are connected via short paths of friendships. Later studies found a similar pattern in online social networks and further extended the theory behind this phenomenon [1, 96,

97]. In this section, I analyze the relationship between shortest paths and local clustering coefficient using 10 real-world social networks. To achieve this, I first introduce the *small-world representation* of a social network and show the small-worldness of the dataset using ω measure. I later proceed to show how local clustering coefficient can be used to estimate the distribution of shortest paths in a network.

3.3.1 Small-World Representation of Data

The focus of this study is to investigate the real-world networks of entities, be it online or physical. The fundamental concept behind small-world is *reachability* of the nodes. Hence, a relationship between nodes that do not enable them to contact or reach each other is not of interest. For example, the connections between individuals who are recipients of the same email do not imply that these users can necessarily reach each other. Reachability through edges in a network can be inferred via the network's *small-world representation*.

Definition 3.4. (Small-world representation). The small-world representation of network $G = (V, E)$ is the undirected network $G' = (V, E')$ such that for all $(v_i, v_j) \in E$, there exists exactly one edge $(v_i, v_j) \in E'$ that represents the flow of information from v_i to v_j and vice versa.

The intuition behind this definition can be better understood from Milgram's broker experiment [79]. In his experiment, Milgram chose a set of individuals at random and asked each of them to send a letter to a specific broker through their connections. Each individual had to choose a person among their acquaintances to pass the letter on. Intuitively, the chosen candidate should have the highest possibility to reach the broker through his/her connections. The flow of information from A to B (i.e. passing the letter from A to B) was entirely dependent on the "acquaintanceship" of A and B. A way to extend this experiment to *virtual societies* (e.g. social media platforms) is through asking someone to pass a message, rumor, or news to a target individual using only their acquaintances.

The acquaintanceship in online networks cannot be defined as straightforward as in physical societies. For example, in an online network like Twitter, one might claim that the *Following* relationship makes a one-sided flow of information from the followee to the follower but not vice versa. I argue that, in terms of the information flow in Milgram's small-world experiment, the flow of information can go from the follower

to followee as well. Consider A, the subject of our experiment, to follow B and B to follow C. If A is asked to pass a message to C through his/her acquaintances, B will be the optimal receiving end of the message despite the fact that B does not follow A. In general, in networks like Twitter, posting content to be seen by one's followers is not the only way of transferring information. Another way is to receive content from the people whom one is followed by in different ways such as *tagging* a person. As a result, I find the small-world representation of a directed social network, such as Twitter, a more reasonable graph model to study paths that deal with information flow. The small-world representation of our example, Twitter network, is its undirected counterpart.

3.3.2 Datasets

Following this strategy, I have selected ten real-world networks with a type of connection among individuals that has a small-world representation. In the following, I introduce each network's type of connection and how they can be modeled as small-world graphs.

- **Zachary's Karate Club [98]:** an undirected network of ties among members of a Karate club after the club splits into two groups.
- **Train Bombing [99]:** an undirected network of contacts among the suspected terrorists in Madrid's train bombing incident in 2004. The original network contains edge weights to show the strength of the connections. However, these weights do not change the *reachability* of the nodes; i.e. an edge between terrorists i and j implies that i can contact j and vice versa regardless of the strength of their relationship (there are no edges with weight zero). So, the small-world representation of the network is the unweighted counterpart of this graph.
- **Residence Hall [100]:** a directed network of friendships among residents of a residence hall on Australian National University campus. A directed edge from i to j shows that i considers j to be a friend. This also implies that i and j know each other whether j considers i as a friend or not. So, the small-world representation will be the undirected counterpart of this graph.
- **Haggle [101]:** an undirected network of contacts between individuals, obtained through carried wireless devices.

- **Infectious [102]:** a multi-edge undirected network of face-to-face contacts among exhibition visitors at Dublin's Science Gallery in 2009. The contacts have been active for at least 20 seconds and multiple contacts could have occurred between two individuals. The small-world representation of this network is the single-edge undirected counterpart.

- **Hamster [103]:** an undirected network of friendships between users in Hamster online social network.

- **Adolescent Health [104]:** a weighted directed network of friendships among students created from a survey in 1994/1995. Each student was asked for the name of his/her top five friends and the edge weights show the frequency of interaction between them. The small-world representation of this network is the unweighted undirected counterpart (same as *residence hall* dataset).

- **Ego Facebook [105]:** an undirected friendship network of Facebook users.

- **Advogato [106]:** a directed network of trust among developers in Advogato platform. The edges have positive weights (the amount of trust between two users) and the nodes can contain self-loops (one can trust himself). The trust between user i and j can imply the prior acquaintanceship between i and j that makes the flow of information possible in both directions. Hence, the small-world representation of this network is the undirected network with no weights or self-loops.

- **Pretty Good Privacy [107]:** the undirected interaction network between users through Pretty Good Privacy (PGP) software.

In all networks, unless otherwise stated, the small-world representation is the same as the original network.

I test the small-worldness of the datasets using the ω measure proposed by Telesford et al. [108]. This measure is defined as

$$\omega = \frac{\mathcal{L}_1}{\mathcal{L}_p} - \frac{C_p}{C_0}. \tag{3.15}$$

The values of ω are in $[-1, 1]$. ω values close to 0 indicate small-worldness (near perfect clustering coefficient and characteristic path length). Negative and positive values show

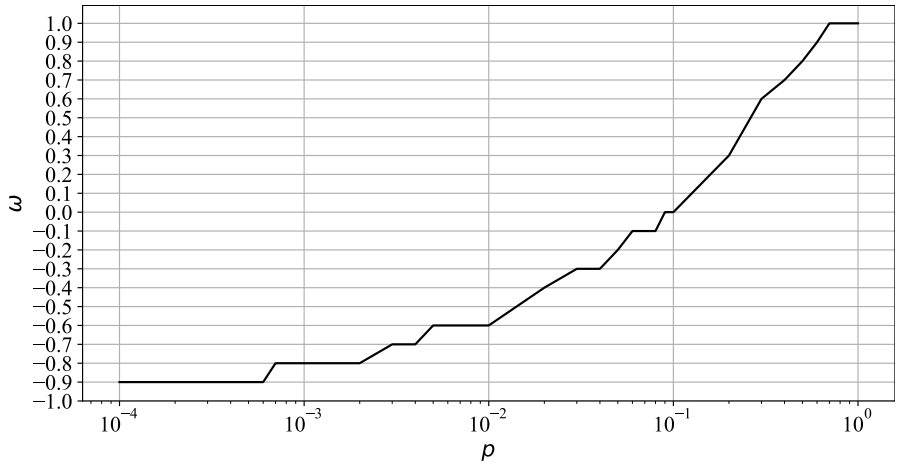


FIGURE 3.5: *Measure of small-worldness, ω, for different values of p in Watts-Strogatz model. The small-world region corresponds to $\omega \in (-0.6, 0]$.*

	edge type	$\|V\|$	$\|E\|$	**avg. deg.**	C **(%)**	$\mathcal{L}$	ω
Zachary Karate Club	tie	34	78	4.59	57.06	2.41	-0.09
Train Bombing	contact	64	243	7.59	62.23	2.69	-0.13
Residence Hall	friendship	217	1,839	16.95	36.28	2.39	-0.09
Haggle	contact	274	2,124	15.5	63.27	2.42	0
Infectious	contact	410	2,765	13.49	45.58	3.63	-0.27
Hamster	friendship	2,000	16,098	16.1	54.01	3.59	-0.18
Adolescent Health	friendship	2,539	10,455	8.24	14.67	4.56	-0.14
Ego Facebook	friendship	2,888	2,981	2.06	2.72	3.87	-0.26
Advogato	trust	5,042	39,227	15.56	25.27	3.27	NA
Pretty Good Privacy	interaction	10,680	24,316	4.55	26.59	7.49	NA

TABLE 3.1: *The network characteristics of 10 real-world datasets. This information belongs to the small-world representation of each network (avg. deg.: average degree).*

more lattice-like and random characteristics, respectively. The corresponding ω for different values of p in Watts-Strogatz model is shown in Figure 3.5. small-world region ($0.01 < p \leq 0.1$) corresponds to $\omega \in (-0.6, 0]$.

The detailed information of these small-world representations can be found in Table 3.1. All datasets (except for Advogato and Pretty Good Privacy that were too large to compute the ω for) have ω values within the small-world region in 3.13 and, hence, are small worlds.

	degree centrality	local clustering coefficient	naïve uniform	uniform	normal
Zachary Karate Club	0.32	0.29	0.14	0.04	**0.03**
Train Bombing	0.44	0.57	0.17	0.09	**0.06**
Residence Hall	0.13	0.09	0.21	0.08	**0.04**
Haggle	1.29	0.65	0.23	0.07	**0.04**
Infectious	0.28	0.19	0.24	0.1	**0.02**
Hamster	0.67	0.19	0.25	0.1	**0.02**
Adolescent Health	0.19	0.35	0.24	0.11	**0.01**
Ego Facebook	1.57	1.82	0.27	0.07	**0.04**
Advogato	1.06	0.32	0.25	0.09	**0.03**
Pretty Good Privacy	0.78	0.19	0.26	0.15	**0.02**

TABLE 3.2: *KL divergence between SPN distribution and that of local information.*

3.3.3 Shortest-Path Distribution and Local Information

In this section, I focus on distributions of local information, i.e. degree distribution, degree centrality distribution, and local clustering coefficient distribution. To infer a meaningful comparison between these distributions and that of shortest paths, I consider the shortest-path distribution of each node. In this distribution, a Shortest-Path Number (SPN) is assigned to each node which is defined in equation 3.18. The sum of SPN index for all nodes in the graph is $|V|$ times the average path length of the graph. For each node i in graph G, the shortest-path number of i is defined as

$$\mathrm{SPN}(i) = \frac{\sum_{j \neq i} d_{\min}(i, j)}{|V| - 1}. \tag{3.16}$$

Note that I assume graph connectivity (bounded SPN). I also used the SPN defined as the median of the shortest paths from i which gave the same results as the average.

The distribution of SPNs for all nodes in the graph (SPN distribution) is of our interest. I use Kullback-Leibler (KL) divergence [109] to measure the distance between SPN distribution and that of local distributions. The KL divergence between two probability distributions P and Q is denoted as $KL(P||Q)$ and means P's divergence from Q. The KL divergence is computed as follows,

$$KL(P||Q) = \sum_{x} P(x) \ln\left(\frac{P(x)}{Q(x)}\right). \tag{3.17}$$

P and Q are identical if $KL(P||Q) = 0$. In our case, P is the distribution of local distribution and Q is that of SPN.

I also test the SPN distribution against a modified version of three standard distributions:

1. **Naïve Uniform**: This distribution models the *random guess* for predicting the SPN of a node. I use this model to test the significance of KL divergence. Any KL divergence value above the corresponding value in naïve uniform model is insignificant. In this model, it is assumed the SPN of each node is drawn from a uniform distribution between the minimum and maximum possible SPN in a graph. Nodes with degree $|V| - 1$ give the minimum possible SPN and maximum SPN occurs if the nodes form a chain such as in $a \rightarrow b \rightarrow c$. In this case, the maximum SPN from equation 3.16 will be

$$\text{SPN}_{\text{max}} = \frac{1 + 2 + ... + |V| - 1}{|V| - 1} = \frac{|V|}{2}. \tag{3.18}$$

   So, the naïve uniform will be defined as $Unif(1, \frac{|V|}{2})$.

2. **Small-World Uniform**: This model is an improvement of random guess. Naive uniform models the true random guess for the SPN of each node with no prior knowledge about the network. However, from small-world phenomenon, we know that the average shortest path from each node is most probably a number less than 10. I use this prior knowledge to make more educated guesses with uniform distribution. I estimate the SPN_{max} as

$$\text{SPN}_{\text{max}} \approx \text{SPN}(n_{mcc}), \tag{3.19}$$

   in which n_{mcc} represents the node with the highest local clustering coefficient (LCC) in the graph. This choice has been made due to (1) the fast calculation of LCC, and (2) the relatively small KL divergence between LCC distribution and SPN (see Table 3.2).

3. **Estimated Normal:** The intuition behind choosing this distribution is the bell shape of the shortest-path length distribution appearing in all of my datasets (Figure 3.6). I use a standard normal distribution which is shifted by SPN_{max} as defined in 3.19.

The KL divergence between all distributions and SPN can be found in Table 3.2. From the table, it is evident that standard normal distribution shifted based on local clustering coefficient measure (see 3.19) models the SPN distribution with the least information loss. This result confirms the theoretical finding for small-world networks; there shortest path length and local clustering information are connected and one can be influenced by changing the other.

3.4 Summary

In this chapter, I studied the relationship between shortest path length (characteristic path length) and local clustering information in small worlds, both theoretically and empirically. I found analytical relationship between the two for Watts-Strogatz model in theory and empirical dependence in real-world small worlds. Considering the importance of shortest paths in formation of contagion paths, these results suggest that by tampering the local clustering information (or local structural community), we can influence the contagion paths and control the magnitude of a spread.

More specifically, the results in the chapter suggest that the best local protection strategy for a critical node is to increase its local clustering coefficient and average shortest path length simultaneously. The former increases the importance of the node in affecting the characteristic path length, and the latter shifts the shortest path distribution such that the characteristic path length is increased.

I use this approach to design algorithms that protect a network against different types of contagion in the next chapters. These algorithms rely on local community information which makes them scalable to larger networks.

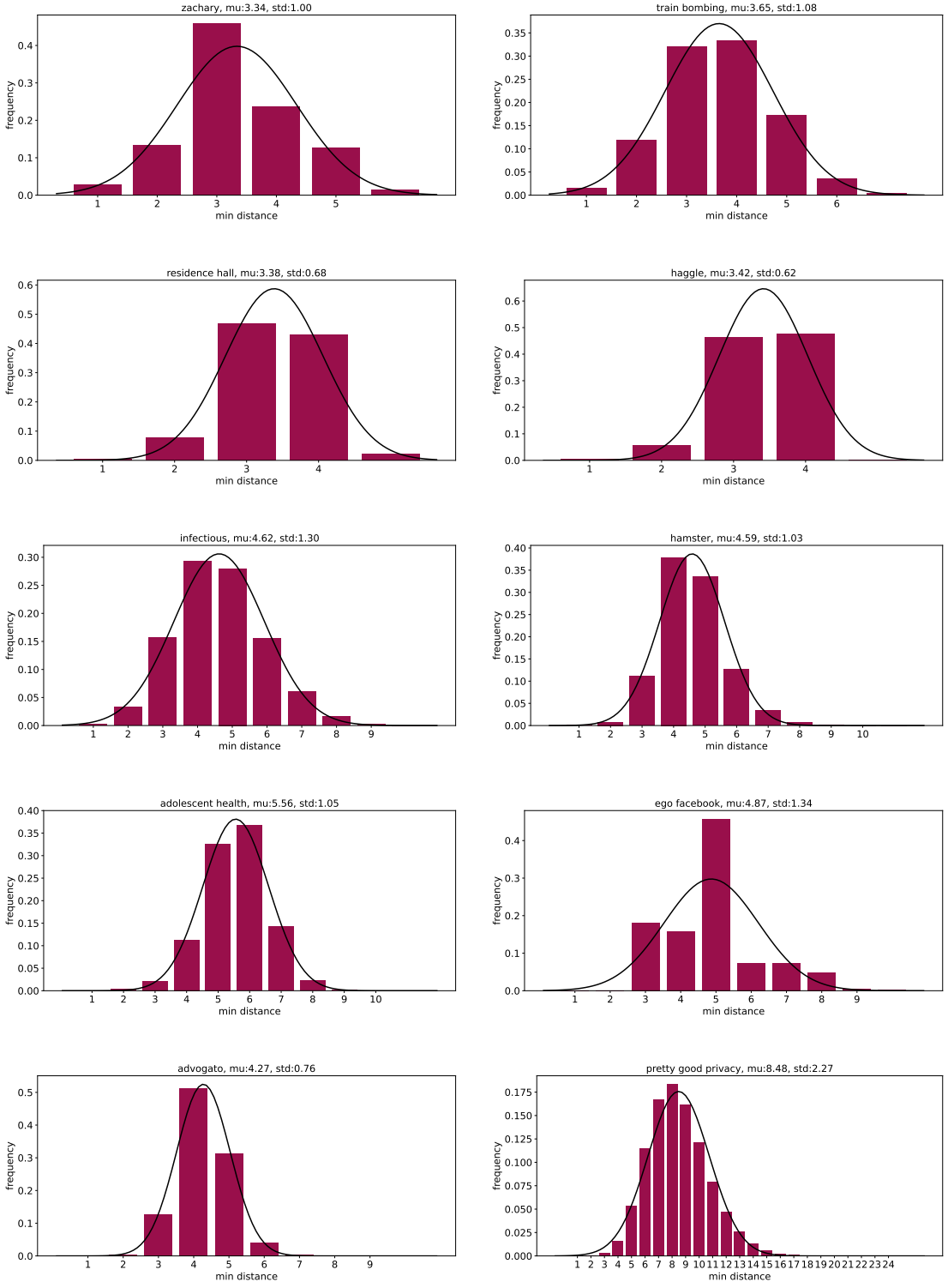


FIGURE 3.6: *The Shortest-path distribution follows a normal distribution.*

Chapter 4

Node Protection against Network Crawling Attacks

With the increase in digitization of entities and their data, the problem of maintaining the privacy of critical nodes in a network has become ever more relevant [110]. Many versions of this problem have been considered. For example, Waniek et al. examine how to protect individuals from detection in a crawling attack [24], and others have addressed the susceptibility of easily accessible public profiles in de-anonymization attempts [111–114]. Numerous studies have been dedicated to finding the optimal crawling strategy for network adversarial attacks and methods for their timely detection [115–119]. This problem can be considered from a variety of angles, including designing effective attack strategies, early detection of attacks, or network defense, in which the goal is to minimally perturb the network so as to protect the target nodes from detection.

In this chapter, I consider the problem of defense against network attack contagions. I assume that the defender has a limited *defense budget* (i.e., number of allowable edge perturbations) for protecting the target nodes, and has no knowledge of the attacker's logistics (i.e., starting point and the crawling algorithm). Note that in this problem, the highest level of protection is achieved by isolating the target. However, this is not an acceptable solution, as these targets are likely to be important to the network, and so removing their connections harms the functionality of the system. The literature contains few works dealing with network protection strategies from the defender's perspective. Previous work has shown that target node protection from the defender's perspective

results is NP-complete [27, 54]. Heuristic solutions use either global graph perturbations [29, 120] or local graph perturbations [24, 26]. The time complexity of the former is substantially greater, and often do not substantially outperform the local methods. However, the search space for local-based methods is small and they rapidly reach their performance plateau regardless of budget (see Section 6.3).

Here, I propose a new approach for vertex defense against crawling attacks: *community-based local graph perturbations*, which find a middle ground between the fast computation of local perturbations and larger search space of global methods.[1] The only information required by *CoVerD* is the community labels of nodes and their 1-hop neighborhood, which is the same information required by local network perturbation heuristics [24]. Therefore, my proposed algorithm is fast and, due to its budget-aware decision making, surpasses the performance of both local and global perturbation heuristics (see Section 4.4). In fact, I show that *CoVerD* has the same impact on reducing the closeness centrality of the target node without the need for expensive computation of centrality. Figure 4.1 depicts the steps of the *CoVerD* algorithm on a toy example. The summary of my contributions in this chapter are as follows,

- I formulate the problem of node protection in complex networks from a defender's perspective. I consider the general case in which the defender has no information on the attacker's starting point or its crawling algorithm.
- I propose a more general heuristic which considers both the local and community information of the target node. My community-based defender, *CoVerD*, is fast and benefits from the advantages of local network perturbations (namely, computational cost and defending budget) while bypassing its shortcomings (namely, the rapid performance plateau). *CoVerD* can achieve close to optimal performance (i.e., the attacker's budget is maximized) and effectively reduces the closeness centrality of the target node.
- On five real-world networks of varying sizes, I show the superiority of *CoVerD* in terms of efficiency and performance against different crawling adversaries.

[1] "Community" here refers to a topological community.

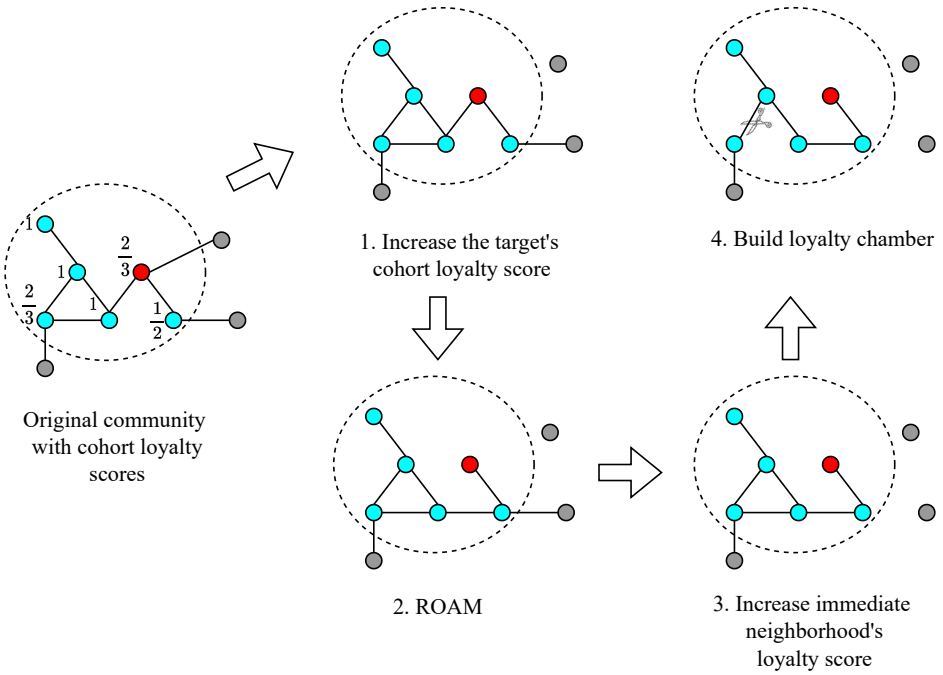


FIGURE 4.1: *CoVerD algorithm on a toy graph. The target nodes and its community members are shown in red and blue, respectively. The nodes outside of the community that are connected to the members are shown in grey. The cohort loyalty scores are included to the left of each member node.*

4.1 Related Work

The problem of defending target nodes against crawling attacks has been studied in four general domains: (1) optimization of crawling techniques for data acquisition (attacker's perspective) [111–114]; (2) detection of malicious crawling behavior (attacker's perspective) [121]; (3) increasing network robustness against crawling attack through global network perturbations (defender's perspective) [27, 29, 54]; (4) protecting target nodes through local network perturbations (defender's perspective) [24, 26]. As the focus is on defender's perspective, here, I only discuss the latter two categories in depth.

4.1.1 Defense via Local Network Perturbations

A simple, yet effective, method in locally manipulating graph structure is ROAM (Remove One, Add Many), the algorithm proposed by [24]. ROAM follows the intuition

that the most important factor in the target node's exposure is its immediate neighborhood. ROAM decreases the degree centrality of the target node by iteratively removing its highest-degree neighbors and connecting them to other immediate neighbors of the target node, ensuring that the average path length and connectivity in the target's neighborhood are preserved.

Abrahamsson adopts the same algorithm but uses eigenvector centrality to pick the neighbor candidate [26]. Their results are comparable to that of ROAM, but incur greater computational costs. The main drawback of ROAM is the limit to its performance. Once the target node's degree reaches 1, ROAM stops and the algorithm reaches its plateau even in the presence of more protection budget. My method matches or beats ROAM's performance for small budgets, but rapidly reaches close to optimal performance with a slight increase in budget. To understand the reason, not that the goal of ROAM is to assure the average path length and connectivity in target's neighborhood is preserved. From the results of Chapter 3, keeping the local average path length preserved while increasing the local clustering coefficient does no effectively change the global characteristic path length. My proposed algorithm escapes this bottleneck by increasing the target's average path length locally via increasing its neighbors' local path lengths.

4.1.2 Defense via Global Network Perturbations

The majority of works in this category use various edge and/or node centralities to greedily remove edges. The objective is to minimize/maximize a global network measure, such as network centrality or average path length. Crescenzi et al. address the complementary problem to ours: maximizing the visibility of a node in the network [27]. They achieve this goal by greedily adding outgoing edges from the target node such that the closeness or betweenness centrality of the target node is maximized. Their time complexity is $O(k.n.g(n,m))$, where n and m are the number of nodes and edges respectively, and $g(n,m)$ is the complexity of computing either closeness or betweenness centrality for a node in the graph.

Numerous works have proposed methods to reduce the complexity of $g(n,m)$ [122–125], among which Ji et al. [29] specifically tailored their method for the vertex protection problem. The time complexity of their approach is $O(k \cdot m \cdot \tau_{mn})$, in which τ_{mn} represents the number of traverse nodes and edges that can be computed in $O(m+n)$ in

the worst case. Despite these efforts, the greedy approach using global network measures do not outperform local measures, such as ROAM [120]) and are infeasible on large scale real-world networks. As such, a few studies have used greedy removal of edges without re-computation of the centrality measures as well [29, 54]. I use three of these methods as baselines that have substantially higher computational cost and worse performance than local methods, including my algorithm, *CoVerD*.

4.2 Preliminaries and Problem Definition

Network Notation. Let $G = (V, E)$ ($|V| = n$ and $|E| = m$) be a connected, undirected, unweighted graph with a mapping $C(.)$ that projects $|V|$ onto non-overlapping partitions. For each node $t \in V$, we represent its 1-hop (immediate) neighborhood in G as $\mathcal{N}_G(t) = \{j | j \in V, (t, j) \in E\}$ and its cohort as $\mathfrak{C}(t) = \{j | j \in V, C(j) = C(t)\}$. The subgraph of G that contains the nodes in $\mathfrak{C}(t)$ is denoted as $\mathcal{G}_t = (\mathfrak{C}(t), \mathcal{E}_t)$, in which $\mathcal{E}_t = \{(i, j) | i, j \in \mathfrak{C}(t), (i, j) \in E\}$. We will refer to this induced subgraph as the node's *cohort subgraph*. Also, the connectivity of the cohort subgraph is the only necessary condition for our algorithm and we can generalize our approach to directed and/or weighted graphs as well (see 4.3.3).

Problem Definition. The vertex defense problem (also referred to as *node protection* and *hiding node* problem [24, 54]) involves a target node t and two actors: a crawling adversary (attacker) and a defender. If we denote the crawling algorithm used by the attacker as $\mathcal{A}$ and the probability of $\mathcal{A}$ visiting a node u in G at the l^{th} step as $P_{\mathcal{A}}(u, G, l)$, the adversary's objective is to find an optimal $\mathcal{A}^*$ within a limited attack budget b_a such that,

$$\begin{aligned} \mathcal{A}^* = \max_{l, \mathcal{A}, b_a} \quad & P_{\mathcal{A}}(t, G, l) \\ \text{s.t.} \quad & l \leq b_a, \quad b_a \leq n. \end{aligned} \tag{4.1}$$

The defender has no knowledge of the attacker's logistics (crawling algorithm, budget, or starting point). It only has information on the target node and the community it belongs to. Within a limited budget $b_d \leq m$, the defender has the ability to perturb any set of edges in the community of t to obtain a new graph G'. Its objective is to find the optimal perturbed graph G^* such that,

$$G^* = \min_{G', b_d} \quad P_{\mathcal{A}}(t, G', l) \qquad (4.2)$$
$$\text{s.t.} \qquad b_d \leq m.$$

4.3 Method

The goal of *CoVerD* is to minimize the closeness centrality of the target node (which increases the attacker's required budget) as much as the available defense budget permits, by using only the information of the cohort neighborhood of the target node. This strategy is intuitive: focus on the immediate neighborhood of the target for small budgets and expand attention to further neighborhoods within the cohort as the budget increases. To this end, *CoVerD* uses a hierarchical modular structure to achieve the proper distribution of the available budget.

There are two underlying assumptions behind *CoVerD*'s intuition: (1) for decreasing the centrality of a node, its 1-hop neighborhood plays a more prominent role than its larger k-hop neighborhood; (2) the existence of a *protective* community structure (i.e., with high average loyalty score) around the target node overpowers the global pathways to the target node. The first assumption is already shown to be the case in real-world social networks [24, 120]. In this study, I intend to show that the second assumption holds for these networks.

4.3.1 Cohort Loyalty Score

While it is tempting to fully isolate a cohort containing a sensitive node, in practice, outgoing connections from cohort are necessary to keep the functionality of the network, even though they increase the cohort's vulnerability. As such, I assign a loyalty score to each node inside of the cohort to signify the impact that a node in the cohort has on exposing the cohort to the rest of the network. Formally, for each node i inside of a cohort $\mathfrak{C}(t)$, its loyalty score with respect to $\mathfrak{C}(t)$ is

$$S_{\mathfrak{C}(t)}(i) = \frac{|\{(i,j)|(i,j) \in \mathcal{E}_t\}|}{|\{(i,j)|(i,j) \in E\}|}. \qquad (4.3)$$

The lower a node's loyalty score, the higher its reach outside of its cohort.

4.3.2 Why Community-based Defense?

The *reachability* of a node is first and foremost defined by its local community neighborhood, as discussed in prior studies in contagion processes [114, 126, 127], which covers the k-hop neighborhood of a target node t for $k = 1, 2, ..., d(t, u)$ with $d(t, u)$ representing the eccentricity of t. For large k, the community's average loyalty score decreases and loses relevance to t. I argue that this is the case in the global target defense algorithms, in which the target node's community structure is ignored (i.e., consideration of global neighborhoods instead of local neighborhood).

The local defense strategies, on the other hand, can be considered another special case of community-based defense in which $k = 1$. However, social networks are shown to have high clusterability in their 2 and 3-hop neighborhoods as well [6, 127], and this is what a community-based method exploits. This gives a balance between capturing a larger search space, without the explosion in computation costs.

4.3.3 CoVerD Algorithm

Figure 4.1 sketches out the *CoVerD* algorithm on a toy graph. *CoVerD* consists of four separate blocks.

The **first block** (Algorithm 3) maximizes the loyalty score of the target node t by removing its connection to neighbors outside of its cohort in the order of those neighbors' degrees. This ordering assures that even for a very limited budget, the target node loses its centrality effectively.

The **second block** (Algorithm 2) is the degree-biased ROAM method [24] that iteratively disconnects the target from its highly connected neighbors. To assure the connectivity, as with the original ROAM algorithm, I make an edge between the disconnected neighbor and one of the immediate neighbors of t. The combination of these two blocks guarantees the high performance of the local perturbations in the absence of sufficient defense budget.

As the budget increases, the third and fourth block boost the performance of the algorithm and break the plateau of the local methods such as ROAM. The **third block**

(Algorithm 4) increases the loyalty score of the $2-hop$ neighborhood of t by removing the $2-hop$ connections that leave $\mathfrak{C}(t)$.

Increasing the loyalty score of this neighborhood promises to have a large impact on decreasing the closeness centrality of the target node (as shown in Figure 4.3). Although the third block achieves a considerable boost compared to using the first two blocks alone, by building a *loyalty chamber* in the target's cohort, I was able to boost the performance of the algorithm for networks with densely connected communities (i.e., a low average loyalty score per community).

CoVerD builds the loyalty chamber in the **fourth block** (Algorithm 5) by using the remaining budget for disconnecting nodes within $\mathfrak{C}(t)$ whose difference in loyalty score is maximum. This last step divides the cohort into two distinctive partitions without disconnecting the graph; the loyal nodes that are purely connected between themselves, and the disloyal nodes that are loosely attached to the cohort. The overall algorithm is shown in Algorithm 1.

4.3.4 Extension to Directed & Weighted Graphs

The connectivity of the target's cohort, $\mathfrak{C}(t)$, is the only necessary condition for running *CoVerD* algorithm. The direction of the edges only impacts the definition of neighborhood $\mathcal{N}_G(t)$ to include the incoming edges only. The edge weights change the definition of loyalty score and budgeting scheme. For a weighted Graph $G = (V, E, W)$, the cohort loyalty score becomes

$$S_{\mathfrak{C}(t)}(i) = \frac{\sum\limits_{j \in \mathcal{N}_{\mathcal{G}_t}(i)} w_{ij}}{\sum\limits_{j \in \mathcal{N}_G(i)} w_{ij}}. \tag{4.4}$$

The budget spent on an edge is equal to the weight of that edge.

4.3.5 Time Complexity

For a target node t, Algorithms 3 and 2 visit at most $|\mathcal{N}_G(t)|$ nodes each. Algorithm 4 iterates over the 2-hop neighborhood of the target and computes the loyalty score for

Algorithm 1: `CoVerD`

Input: G, t, $\mathfrak{C}(t)$, b_d
continue $\leftarrow$ *True*
while *continue* **do**
    $G, b_d, \textit{continue} \leftarrow$ `IncreaseTargetLoyalty(`G, t, $\mathfrak{C}(t)$, b_d`)`
    $N \leftarrow \mathcal{N}_{\mathcal{G}_t}(t)$
    $G, b_d, \textit{continue} \leftarrow$ `ROAM(`G, t, b_d`)`
    $G, b_d, \textit{continue} \leftarrow$ `Increase1HopLoyalty(`G, N, t, $\mathfrak{C}(t)$, b_d`)`
    $G, b_d, \textit{continue} \leftarrow$ `BuildLoyaltyChamber(`G, t, $\mathfrak{C}(t)$, b_d`)`
end
return G

each node in the cohort, visiting on average $|\mathcal{N}_G(t)| \cdot (\overline{|\mathcal{N}_G|} + |\mathfrak{C}_t|)$, in which the $\overline{|\mathcal{N}_G|}$ is the average degree in G. Algorithm 5 visits every node in the cohort once to re-compute their loyalty score and visits exactly $|\mathfrak{C}_t|$ nodes. For the average case in which $|\mathcal{N}_G(t)| \approx \overline{|\mathcal{N}_G|} = \mathfrak{d}$, the overall time complexity of *CoVerD* is $\mathcal{O}((\mathfrak{d}+1) \cdot |\mathfrak{C}_t| + \mathfrak{d}^2 + 2\mathfrak{d})$, which for $\mathfrak{d} \ll |\mathfrak{C}_t|$ curtails to $\mathcal{O}(\mathfrak{d} \cdot |\mathfrak{C}_t|)$. So, the speed of *CoVerD* depends mainly on the size of the target's community. This is a significant improvement from the polynomial time complexity of global methods (see Section 6.1) and is still comparable to local methods with average time complexity of $\mathcal{O}(\mathfrak{d})$.

Algorithm 2: `ROAM`

Input: G, t, b_d

$flag \leftarrow False$

$spent \leftarrow 0$

while $spent \leq b_d$ **do**
    $\mathcal{N}_{\mathcal{G}_t}(t) \leftarrow \texttt{SortByDegree}(\mathcal{N}_{\mathcal{G}_t}(t))$
    for $p \in \mathcal{N}_{\mathcal{G}_t}(t)$ **do**
        $G' \leftarrow G(V, E \setminus \{(t, p)\})$
        if $\texttt{IsConnected}(G')$ **then**
            $G \leftarrow G'$
            $q \leftarrow \texttt{Random}(\mathcal{N}_{\mathcal{G}_t}(t))$
            $G \leftarrow G(V, E \cup \{(p, q)\})$
            $spent \leftarrow spent + 2$
        end
    end
end

$b_d \leftarrow b_d - spent$

if $b_d \leq 0$ **then**
    $flag \leftarrow True$
end

return $G, b_d, flag$

Algorithm 3: `IncreaseTargetLoyalty`

Input: G, t, $\mathfrak{C}(t)$, b_d

$flag \leftarrow True$

$spent \leftarrow 0$

$N' \leftarrow$ `SortByDegree`$(\mathcal{N}_G(t) \setminus \mathcal{N}_{\mathcal{G}_t}(t))$

for $p \in N'$ **do**
    $G' \leftarrow G(V, E \setminus \{(t, p)\})$
    if `IsConnected`(G') **then**
        $G \leftarrow G'$
        $spent \leftarrow spent + 1$
        if $spent > b_d$ **then**
            break
        end
    end
end

$b_d \leftarrow b_d - spent$

if $b_d \leq 0$ **then**
    $flag \leftarrow False$
end

return $G, b_d, flag$

Algorithm 4: `Increase1HopLoyalty`

Input: $G, N, t, \mathfrak{C}(t), b_d$

$flag \leftarrow True$; $spent \leftarrow 0$

for $p \in N$ **do**
  $N' \leftarrow$ `SortByDegree`$(\mathcal{N}_G(p) \setminus \mathcal{N}_{\mathcal{G}_t}(p))$
  for $q \in N'$ **do**
    $G' \leftarrow G(V, E \setminus \{(p,q)\})$
    if `IsConnected`(G') **then**
      $G \leftarrow G'$; $spent \leftarrow spent + 1$
      if $spent > b_d$ **then**
        break
      end
    end
  end
  if $spent > b_d$ **then**
    break
  end
  for $q \in \mathfrak{C}(t)$ **do**
    Compute $S_{\mathfrak{C}(t)}(q)$
  end
  for $q \in \mathcal{N}_{\mathcal{G}_t}(p)$ **do**
    if $S_{\mathfrak{C}(t)}(q) < 1$ **then**
      $G' \leftarrow G(V, E \setminus \{(p,q)\})$
      if `IsConnected`(G') **then**
        $G \leftarrow G'$; $spent \leftarrow spent + 1$
        if $spent > b_d$ **then**
          break
        end
      end
    end
  end
  if $spent > b_d$ **then**
    break
  end
end

$b_d \leftarrow b_d - spent$

if $b_d \leq 0$ **then**
  $flag \leftarrow False$
end

return $G, b_d, flag$

Algorithm 5: `BuildLoyaltyChamber`

Input: G, t, $\mathfrak{C}(t)$, b_d

$flag \leftarrow True$; $spent \leftarrow 0$

for $p \in \mathfrak{C}(t)$ **do**
  compute $S_{\mathfrak{C}(t)}(p)$
end

$candid \leftarrow \{(n_1, n_2) | (n_1, n_2) \in \mathcal{E}_t, S_{\mathfrak{C}(t)}(n_1) = 1, S_{\mathfrak{C}(t)}(n_2) < 1\}$

$candid \leftarrow$`SortByScoreDiff`$(candid)$

for $q \in candid$ **do**
  for $p \in \mathcal{N}_{\mathcal{G}_t}(q)$ **do**
    $G' \leftarrow G(V, E \setminus \{(p, q)\})$
    if `IsConnected`(G') **then**
      $G \leftarrow G'$; $spent \leftarrow spent + 1$
      if $spent > b_d$ **then**
        break
      end
    end
  end
end

$b_d \leftarrow b_d - spent$

if $b_d \leq 0$ **then**
  $flag \leftarrow False$
end

return $G, b_d, flag$

4.4 Experiments

In this section, I analyze the performance of *CoVerD* algorithm against both local and global perturbation algorithms on five real-world datasets.

4.4.1 Experimental Setup

In my experiments, I implement a defense strategy (edge perturbations) on a given network G for a target node t to obtain *defended* network G^*. Then, I run a crawling algorithm starting from a given source node and obtain b_a, the number of nodes explored by the adversary crawler, before reaching t. b_a is at most $|V|$, so the defender performance metric is $\frac{b_a}{|V|}$. I select target nodes in three ways, and for each, select 5 nodes:

- **Random Targets:** The target nodes are chosen uniformly at random.
- **Degree-based Targets:** The targets are chosen with probability proportional to degree. This strategy mirrors the attack on well-connected influential nodes.
- **Community-based Targets:** First, each community receives two scores, each in $[0, 1]$, based on (a) their size and (b) density of their intra-group edges. The normalized sum of these two scores gives a final ranking of each community. The targets are chosen from $|V|$ with a probability that is biased towards the score of their respective community. This strategy mirrors the attack on well-connected influential communities.

I choose five different source nodes at random for the attacker's starting point. I run the simulation for all (target, source) combinations (i.e., 75 pairs) and report their average performance. This procedure is repeated for values of b_d ranging from 0.1% to 5% of the edges in the graph.

Datasets. I use five real-world datasets whose names and basic statistics are shown in Table 4.1. For the community assignment in each network, I use Louvain community detection method [14].

Defender algorithms. I compare my algorithm against both local and global defenders. ROAM is the most prominent local-perturbation method [24] (see 6.1). Among the global perturbations, however, my choices are limited to those that have feasible

computation time on the selected real-world networks. I build three global defenders by following the proposed approximate global perturbation methods in [54, 116]. In my four baselines: **ROAM** is implemented similarly to Algorithm 2, except that I do not limit the neighborhood of the target to the cohort neighbors. **Betweenness, PageRank**, and **MaxDegree** score each edge as the sum of it's endpoints' scores, where each node is scored by its betweenness centrality, PageRank, or MaxDegree, respectively. The top scoring b_d edges are removed.

Attacker algorithms. According to [121], the hallmark of aggressive crawling is the choice of an expansion-based method that allows for as far as possible from the starting point, such as depth-first search (DFS). On the other hand, innocent crawlers tend to remain in the local neighborhood of the starting node, and tend to resemble breadth-first search (BFS). As such, I have chosen these two crawling techniques to show the performance of my algorithm in the presence of both aggressive and innocent crawlers. (Note, however, that *CoVerD* is agnostic to the crawling algorithm.)

4.4.2 Results

Table 4.1 shows the result[2] of my experiments for $b_d = 0.006|V|$. Against the BFS crawler, *CoVerD* shows a pronounced superior performance and, in some instances, it increases the crawler's budget by 3 to 10 times compared to the next best-performing benchmark (see the result for degree-based target node of `deezer` and `github`). In 60% of simulations, it achieves close to perfect results ($b_a \geq 0.96$).

The hardest dataset for this task was `twtich`, for which the results of the best performing models, *CoVerD* and ROAM, are relatively low. However, compared to the undefended graph and the three global methods, *CoVerD* still increases the attacker's budget by $\approx 50\%$. Against the more aggressive crawling scheme of DFS, all models perform worse than their BFS counterpart, as expected. Nonetheless, *CoVerD* outperforms all benchmarks in 80% of simulations and, in the remaining cases, it offers competitive results.

Figure 4.2 shows the change in defenders' performance with respect to their availbe budget. In all cases, *CoVerD* reaches near optimal performance with budgets less than $0.01|V|$. ROAM also offers decent results in the majority of the cases. However, its

[2]For the largest dataset, `github`, obtaining the results of the global measures was infeasible

Networks	Defender	Random Target		Degree-based Target		Community-based Target	
		BFS	DFS	BFS	DFS	BFS	DFS
lastfm-asia $\|V\| = 7,624$ $\|E\| = 27,806$	Original	0.12	0.28	0.54	0.20	0.48	0.21
	Betweenness	0.16	0.38	0.53	0.27	0.49	0.37
	PageRank	0.23	0.23	0.55	0.19	0.5	0.34
	MaxDegree	0.18	0.38	0.55	0.27	0.43	0.33
	ROAM	0.44	0.55	0.90	**0.58**	0.84	0.40
	CoVerD	**0.99**	**0.68**	**1.00**	0.56	**0.96**	**0.50**
musae-twitch $\|V\| = 7,126$ $\|E\| = 35,324$	Original	0.19	0.86	0.35	0.18	0.53	0.43
	Betweenness	0.20	0.88	0.31	0.17	0.55	**0.69**
	PageRank	0.26	0.88	0.31	0.20	0.55	0.47
	MaxDegree	0.31	0.69	0.30	0.18	0.54	0.37
	ROAM	0.45	0.86	0.84	0.29	**0.89**	0.58
	CoVerD	**0.48**	**0.94**	**0.95**	**0.78**	**0.89**	0.53
deezer-europe $\|V\| = 28,281$ $\|E\| = 92,752$	Original	0.20	0.26	0.24	0.57	0.12	0.22
	Betweenness	0.23	0.18	0.33	0.67	0.11	0.16
	PageRank	0.19	0.18	0.23	0.54	0.11	0.23
	MaxDegree	0.19	0.16	0.23	0.39	0.12	0.15
	ROAM	0.56	0.52	0.23	0.39	0.42	0.52
	CoVerD	**0.82**	**0.56**	**0.92**	**0.93**	**0.99**	**0.79**
musae-facebook $\|V\| = 22,470$ $\|E\| = 171,002$	Original	0.32	0.38	0.49	0.39	0.8	0.69
	Betweenness	0.33	0.36	0.49	0.37	0.76	0.68
	PageRank	0.31	0.51	0.51	0.33	0.80	0.70
	MaxDegree	0.33	0.17	0.56	0.40	0.79	0.59
	ROAM	0.85	**0.71**	0.83	**0.65**	0.80	0.58
	CoVerD	**0.99**	0.65	**0.98**	**0.65**	**0.98**	**0.89**
musae-github $\|V\| = 37,700$ $\|E\| = 289,003$	Original	0.27	0.35	0.00	0.01	0.33	0.18
	Betweenness	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*
	PageRank	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*
	MaxDegree	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*
	ROAM	0.93	0.52	0.46	0.05	0.90	0.49
	CoVerD	**0.98**	**0.64**	**0.86**	**0.59**	**0.97**	**0.53**

TABLE 4.1: *The performance of all defenders against BFS and DFS crawling attacks for different types of target nodes. The values show the normalized attacker budget ($\frac{b_a}{|V|}$) in order to discover the target node. The values closer to* 1 *indicate superior performance of the defender and are shown in* ***bold****. For BFS crawlers, CoVerD always surpasses the benchmarks with considerable Margie. The same holds true for DFS crawlers in the majority of cases. In general, all defenders perform worse against the DFS crawling attack (aggressive crawling).*

effectiveness rapidly reaches its plateau and never offers a near-optimal result. Its surprisingly poor performance for `deezer` in Figure 4.2, in contrast to the near-optimal performance of **CoverD**, suggests the importance of looking beyond the immediate neighborhood of the target node. Recall that for small defense budgets, the only difference between *CoVerD* and ROAM is the maximization of target's loyalty score, $S_{\mathfrak{C}(t)}(t)$. Hence, the superior performance of *CoVerD* for small budgets versus that of ROAM in Figure 4.2 shows the importance of community membership in determining a node's reachability.

In all previous studies, the indicator of a defender's success was defined by its ability

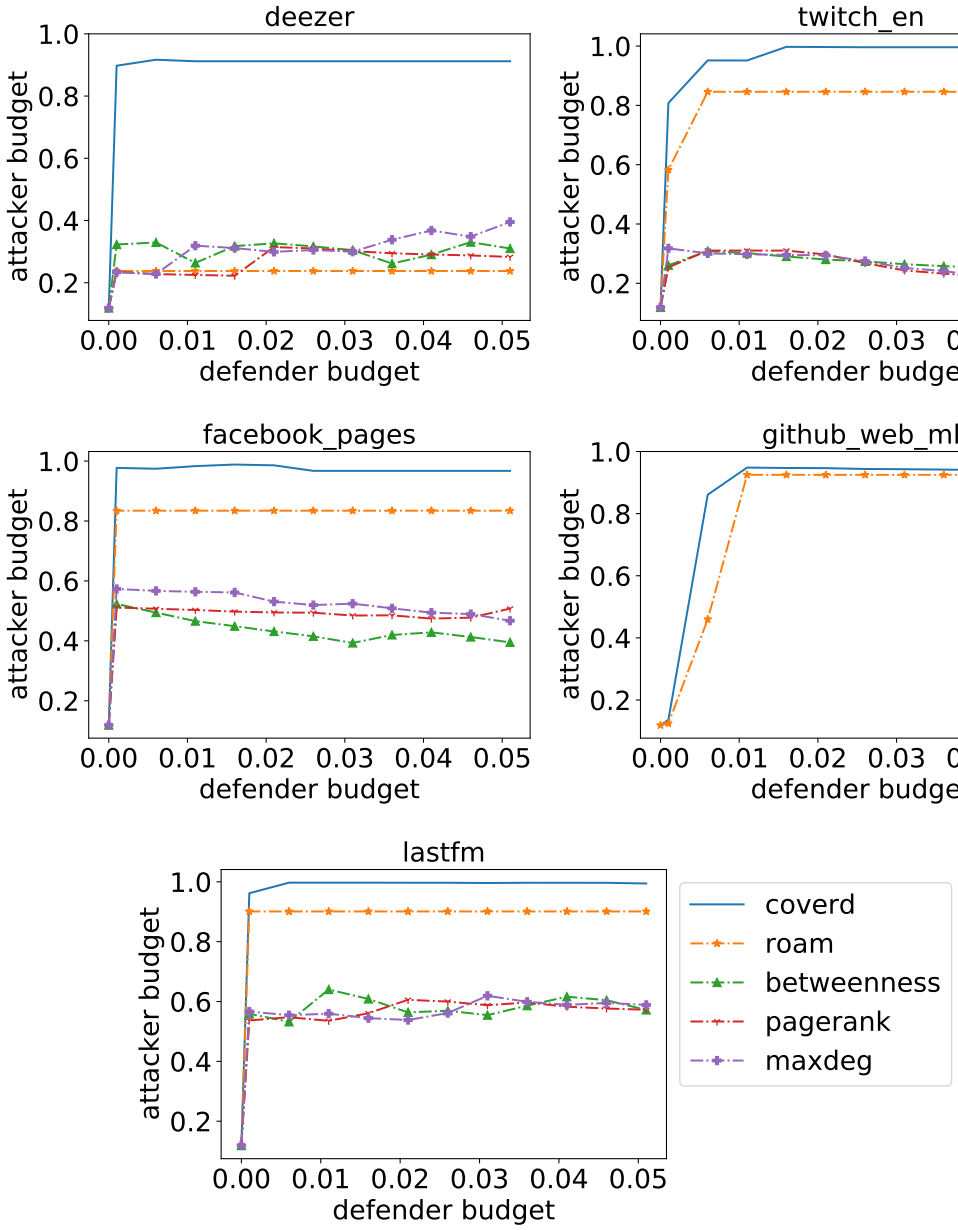


FIGURE 4.2: *The defender budget vs. attacker budget for different defender algorithms. The plots show the aggregated simulation results for degree-based target nodes and BFS crawling attack. Similar results were obtained for DFS attack as well as community-based and random target nodes. CoVerD outperforms all the baselines for the same values of b_d. It also reaches the optimal performance ($b_a \approx 1$) on the majority of datasets.*

to minimize the closeness (or betweenness) centrality of a target node (in contrast to ours in which the increase in b_a marks the performance). I also show the change in the closeness centrality of the target nodes for different b_d in figure 4.3. Even though I did not use any global measures to decrease the closeness centrality directly, *CoVerD* has achieved the fastest and deepest drop in the target's centrality by focusing only on its local community structure. This figure also shows that for achieving comparable

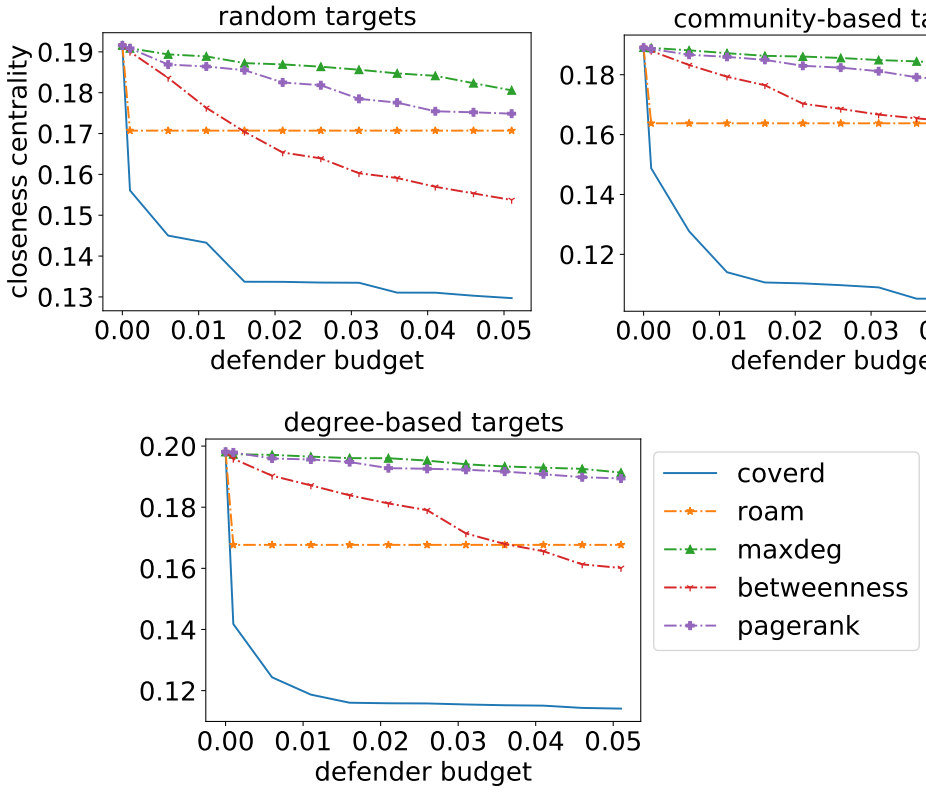


FIGURE 4.3: *Closeness centrality of the target node (y-axis) vs. the defense budget (x-axis). The plots belong to* `lastfm` *data. For each plot, I have used the mean of the closeness centrality among all the target nodes. It is evident that CoVerD substantially surpasses both local and global measures in decreasing the closeness centrality of the targets for all target types.*

performance with [already computationally expensive] global defenders, such as the Betweenness model, I need to invest in larger defense budgets (note the slow but steady decrease of the centrality for Betweenness model in Figure 4.3).

4.5 Summary

In this chapter, I formalized the problem of vertex protection from a defender's perspective. I proposed the *CoVerD* heuristic that leverages the community structure of social networks. This algorithm retains the fast computation of local network perturbations and shows superior performance compared to both local and global defenders. Despite using only the local community information, my algorithm achieves a substantially lower closeness centrality than both local and global perturbation models.

Future Direction. This study is an important step forward in the field of network protection and privacy to focus on heuristics that are both practical and efficient in the real-world settings. Two valuable extensions to *CoVerD* are (1) investigating the correlation between different community structures and defender's performance; (2) introducing additional constraints to the defender's decision making, such as maintaining certain properties of the network or avoiding the formation of certain motif subgraphs.

Chapter 5

Early Mitigation Strategies against Viral Spread

In response to recent pandemics, such as the COVID-19, governments across the world have attempted a variety of strategies to mitigate the spread of disease, including lockdowns, contact tracing, and others. However, there has been little analysis on the relative merits of such strategies; and because these strategies have negative effects on the economy and the morale of the people, it is critically important to understand their efficacy. Although the topic of this chapter is inspired by the COVID-19 pandemic, my analysis can apply to any contagious disease. Because the response to a pandemic depends on whether it is in the early stages (no vaccination available) or later stages (vaccination available), I focus on early pandemic mitigation strategies in correspondence with the COVID-19 situation.

In this chapter, I analyze variants of the pandemic mitigation strategies practiced in the real world – e.g. lockdown and test-trace-isolate – from a network perspective. Inspired by the new psychology findings on the correlation between community membership and pandemic response [128–130], I also show that the mitigation strategy becomes even more powerful if we incorporate the community information, as expected from the results in Chapter 4. The goal of this analysis is to capture the efficacy of the current protection schemes against viral contagion and draw on their merit and shortcomings. I will use these results in the next chapter where I address a more challenging protection scenario for which no current algorithm exists.

To evaluate each protection (mitigation) strategy, unlike the majority of related work in this field, I consider both the magnitude of spread and economic impact as cost factors.

For example, a mitigation strategy such as a total lockdown might have the best performance in terms of controlling the spread of disease if prolonged for long enough time until the discovery of the vaccine. However, the devastating economic impacts of such a decision makes this strategy inefficient in real world.

On the other hand, the "Do Nothing" strategy, which relies on herd immunity (see Section 5.2), results in less economic impact (at least in the early stages), but does nothing about the spread of the disease. An ideal mitigation strategy should offer a trade-off between these two losses. I allocate a budget to each strategy to count for the economic impact and report both the spread and budget spent for each strategy simulation.

To ensure that the results are generalizable to other contagious diseases, I enforce only general assumptions about the nature of the disease and cost of battling the spread (see Section 6.2). I use the SIRD epidemic model for the simulations and only consider the budget spent on isolation strategies. I validate each strategy on a set of 10 real-world social networks (see Section 5.3.2). To have a close approximation of human-human contact behavior, these networks are chosen based on the method of data collection and the meaning of connections between two individuals. I also consider a set of online social networks that are frequently used in the disease spread literature [131–133] and, in some cases, have been shown to give a close-enough approximation of real-world social networks [132].

My results show the superiority of the test-trace-isolation strategy if combined with k-hop neighborhood ranking (specifically for k = 1). I also confirm the theoretical results from psychology studies on the impact of community membership in reducing the spread of the disease and show the further direction in adopting such strategies.

5.1 Problem Statement

I model the population as a simple undirected graph $G(V, E)$, where individuals are represented by nodes (V) and connections between them (E) represent physical contact. I use undirected edges due to the nature of physical contact, for which a directed relationship does not bear any meaning. I also consider unweighted and un-attributed graphs, as attribute information is not easy to gather in a real-world setting and in the practical strategies discussed below. However, the simulations can easily be extended to attributed or weighted graphs. For example, the strength of an edge can be considered

as the frequency or length of the contact, where a higher value increases the probability of infection spread. In the following discussion, I discuss different models of disease spread and the reasons behind my choice of the SIRD model. I will also discuss my method of extending the model to count for the economic impact.

5.1.1 Viral Spread Modeling

Previous work on mathematical modeling of viral spread can be grouped into two categories of (1) general spread models and (2) virus-specific spread models. The former includes famous models such as SIR, SIRD, SIS, SIER, and SIRS [134–136]. The virus-specific models have been proposed for viruses observed in real world and consider the specific properties of a certain virus into the modeling of the spread [137–139].

The focus of this chapter is on the effectiveness of different mitigation strategies for an unknown pandemic scenario (i.e., a pandemic whose specific behavior and potential remedies are unknown). It is known that battling a new pandemic heavily relies on adopting a proper mitigation strategy in its early stages [140]. In these early stages, our knowledge of the nature of the virus is very limited, and the virus-specific strategies require prior knowledge gained from time-consuming clinical trials. Thus, the general models with little to no conditions on virus-specific behavior are more practically applicable in the early stages of a pandemic. For my analysis in this section, I choose the SIRD model due to its minimal assumptions on the nature of a fatal spread, which is explained below.

5.1.2 SIRD Epidemic Model

Given a *closed* community in which the population is fixed (no birth, no migration, and no death due to causes irrelevant to the disease under study), the SIRD model assumes four possible states for each individual in the community at each timestamp: Suceptible (never contaminated by the virus), Infectious (contaminated and can spread the virus), Recovered (recovered from contamination and can no longer spread the virus), and Dead (due to infection). The possible transition between states and their respective probabilities are depicted in figure 5.1.

The three parameters of this model are α, β, and γ that indicate infection, recovery, and mortality rate respectively. Their exact values used in the simulations are presented in

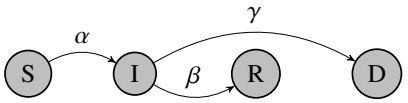


FIGURE 5.1: *SIRD state transitions. Parameters α, β, and γ indicate infection, recovery, and mortality rate respectively.*

Table 5.2 and discussed in Section 6.4. In Section 5.5, I discuss the choice of these hyperparameters and validate the robustness of the results for different values of α, β, and γ. I use a discrete-time SIRD model with discretization period of duration one day. Having initial values S_0, I_0, R_0, D_0 in a population of size N, the virus spread follows these laws of motion:

$$S_{t+1} = S_t - \alpha \frac{I_t S_t}{S_t + I_t} \tag{5.1}$$

$$I_{t+1} = I_t + \alpha \frac{I_t S_t}{S_t + I_t} - (\beta + \gamma) I_t \tag{5.2}$$

$$R_{t+1} = R_t + \beta I_t \tag{5.3}$$

$$D_{t+1} = D_t + \gamma I_t \tag{5.4}$$

I assume that no individual can stay in I state indefinitely. As such, every infectious individual can only stay infected for a certain amount of time (disease duration in Table 5.2) and transitions to R if not deceased or recovered already. Note that I do not consider any delay in the transitions.

5.1.3 Budget Allocation

The exact modeling of a pandemic's economic impact is a complicated problem and requires a comprehensive study on its own [141–143]. However, we still can introduce a simplified measure of cost for comparison between different mitigation strategies. As we try to minimize the number of isolated individuals while reducing the rate of spread, we have to compensate for the portion of the population that is under quarantine (either compulsorily or voluntarily) to make isolation practical and possible without threatening the well-being of families and individuals.

I treat this compensation as a required budget for each isolation strategy. Ultimately, an ideal isolation strategy should use a small compensation budget while minimizing the

peak number of the infectious population over time. A smaller amount of budget spent also indicates isolated individuals, implying less possibility of economic impact due to work-force perturbation.

5.2 Mitigation Strategies

Two of the most important problems in the early stages of a pandemic are (1) the capacity of healthcare centers and (2) economic consequences [141–144]. An optimal mitigation strategy seeks to reduce the occupancy of hospitals (lower the number of infected) while maintaining the productivity of the society to eliminate economic impacts. However, these two objectives often bear conflicting interests.

So far, the strategies for lowering the number of infected individuals practiced in real-world setting have negatively affected the economic well-being of the society. A current example is the **Lockdown strategy** adopted by many countries (such as the USA, Spain, and Italy) in 2020 to mitigate the COVID-19 spread[1]. Interestingly, lockdown does not offer an optimal solution to either of the objectives above. First, lockdown leads to a *second wave* of spread and has to be implemented in several phases to be effective in lowering the burden on the healthcare system [145]. Second, it is shown (both in theory and practice) that lockdown strategy causes severe damages to the economy [144].

To trade-off between the need for isolation and economic prosperity, [144] suggests employing a **Test-Trace-Isolate strategy (TTI)**. This method puts the focus on the neighborhood of the individuals with positive test results (infected). According to [146], the countries who employed the TTI strategy against COVID-19 were able to combat the spread more successfully than those who followed *herd immunity*[2] or full containment (lockdown) strategy. This, however, mainly considers the medical benefits of the mitigation. The cost-effectiveness of TTI (economical aspect) heavily depends on its implementation [147]. For example, how do we choose whose neighborhood to trace? Is it the people who show symptoms or those who have tested positive? Furthermore, how many people in the candidate's neighborhood should we isolate and how big should the size of this neighborhood be?

[1] https://en.wikipedia.org/wiki/COVID-19_pandemic_lockdowns

[2] Herd Immunity is an epidemiological concept and is defined as "the percentage of people with protective immunity needed in a population to stop the propagation of an infectious agent" [146]. This strategy, although seemingly giving an optimal solution to economic impact of the spread, results in a devastating death toll in the population.

Tracing and isolating steps of TTI are costly and, if implemented in a naïve way, it can be less efficient than lockdown strategy. Here, I examine three different strategies for TTI. These methods all use local neighborhood information. I consider random and centrality-based TTI with tracing radius up to k-hops away from the infected node (for $k \in \{1, 2\}$). The details of each method are discussed in Section 6.4. Note that due to small-world property of social networks, for k values higher than two, we capture almost all of whole network, which is counter-intuitive for TTI strategy.

In a pandemic, human behavior plays as important of a role as properties of the virus (if not a more important role) [128]. Such behavior is directly connected to psychological traits of individual's personality [129]. An interesting relevant observation is that shared community membership increases the speed of the spread [130]. Previous studies have shown there is a correlation between community structure and spread behavior (e.g., under certain community structure, the spread is slowed down or sped up) [61, 148]. In a recent study, [149] shows that community size and density play an important role in the predictability and controllablity of epidemic.

These observations are used in immunization literature to leverage the community structure for optimizing the immunization plan. For example, in [25], authors propose a heuristic for finding potential community bridges and immunize them. In [126], several ranking methods based on the in/out degree of nodes in a community are proposed to choose a community for immunization.

The immunization is mainly done when enough information is available about the virus and it is possible to use time-consuming heuristics for finding the optimized set of nodes to immunize. Moreover, the immunization is not as costly as isolation strategies due to the prolonged nature of the latter.The target of current study is to act in the early-stages and with limited to no knowledge on the nature of the disease. We tend to find a balance between lowering the cost and the peak of infection by using isolation strategies in the absence of vaccines/remedies. I argue that by considering the findings in the aforementioned studies, we can improve isolation-based mitigation strategies.

As such, I propose a **Community-based Isolation strategy (CI)** and show its effectiveness in comparison to lockdown and TTI strategies. The results of CI are presented to show that using the community membership of individuals as an isolation strategy indeed reduces the speed of spread. At first glance, this method might not be as practical as lockdown or TTI, owing to the fact that community membership is only partially known and tracing the memberships can be even more costly that TTI (as shown in

Section 6.4). However, the experiment results show that community-based isolation surpasses all other methods in reducing the spread of the disease without the disadvantage of a second wave.

5.3 Experimental Setup

In this section, I evaluate each of the baseline strategies against the proposed methods. I focus only on real-world data to consider the complex dynamics of the human-human interaction, which is not entirely captured in synthetic networks (e.g, stochastic block model, small-world model, etc.).

5.3.1 Assumptions

As mentioned before, for lockdown and TTI, I do not enforce any disease-specific information on the model. Additionally, I do not assume the presence of network structural data that are hard or impossible to obtain in real-world setting. For example, I do not assume that we have global information for nodes or edges (e.g. shortest path-based centralities, diameter of the network, or spectral properties). We only have the information on the neighborhood of each individual (as obtained through individual surveys in real world).

For CI, I assume the community membership of individuals is known. In the real-world, the communities can be considered at different levels; from a club membership level up to county and state levels. Considering that not all of the datasets have ground-truth communities, I obtain membership through Louvain partitioning of the graph that maximizes the modularity.

5.3.2 Data

Recent studies show the importance of using real-world human-human interaction data to account for the influence of human behavior in the simulation of a spread [128]. I chose seven real-world datasets that have been collected based on physical human interaction/connections in real world. These datasets, although the best resource for real-world interactions, are generally small due to the cost of data collection. As such,

Data	Type	Edge Meaning	$\|V\|$	$\|E\|$	Avg. Deg.
Infectious (INF)	Human Interaction	Contact	410	2,765	13.49
Hyptertext2009 (HX9)	Human Interaction	Contact	113	2,196	38.87
Haggle (HAG)	Human Interaction	Contact	315	2,899	18.41
Adolescent Health (AH)	Human Social	Friendship	2,539	10,455	8.23
Residence Hall (RH)	Human Social	Friendship	217	1,839	16.95
Physicians (PHY)	Human Social	Trust	241	923	7.66
Jazz Musicians (JAZ)	Human Social	Collaboration	198	2,741	27.69
Pretty Good Privacy (PGP)	Online Contact	Interaction	10,680	24,316	4.55
Facebook NIPS (FBN)	Online Social	Friendship	2,888	2,981	2.06
Hamster Full (HAM)	Online Social	Friendship	2,426	16,630	13.71

TABLE 5.1: *Contact datasets for spread simulation*

many studies tend to use online social networks as approximate behavior of the users in physical world. [132] showed that online behavior approximation for some online networks such as Facebook is close to physical behavior. To both confirm their results for other online social networks and consider the simulation results on larger networks, I also consider an additional three larger datasets from online social networks. All of the 10 datasets are chosen based on the nature of their contact (edge meaning). These datasets and their general statistics are shown in Table 6.2. All datasets are publicly available in the Konect repository [150].

5.3.3 Implementation of Mitigation Strategies

In this section, I briefly go over the implementation of each mitigation strategy mentioned in Section 5.2. The hyperparameters are the same among all strategies (such as quarantine compensation, duration of quarantine, duration of disease, and parameters α, β, γ). These hyperparameters are presented in Table 5.2. In all simulations, I start with only one infectious node chosen at random (i.e, $I_0 = 1$, $S_0 = |V| - 1$, $R_0 = D_0 = 0$, unless otherwise specified). For each model, I repeat the simulation for 100 different starting node and report the average among the 100 trials. Each round of simulation is run until there are no infected nodes left in the network. In each timestamp t_i, the number of infected, susceptible, and quarantined individuals are reported based on the networks status in t_i. The reported proportion of infected individuals is averaged over the timestamps in the simulation. The number of deceased and recovered individuals are reported cumulatively (from t_0 to t_i).

- **Do Nothing (DN)**: Although not exactly a mitigation strategy, DN can be used as the baseline to compare the performance of other methods against it. It is a simple

Hyperparameter	Value
Probability of infection (α)	0.2
Probability of recovery (β)	0.08
Probability of death (γ)	0.04
Simulation unit of time	1 day
Disease duration	7 days
Quarantine duration	14 days
Daily quarantine compensation (Compulsory Isolation)	\$100
Daily quarantine compensation (Volunteer isolation)	\$50
Volunteer quarantine probability	0.5

TABLE 5.2: *Hyperparameters chosen for all mitigation strategies when applicable.*

SIRD model (Equation 5.1 – 5.4) that reaches the peak of infection quickly and fades away quickly as well (due to herd immunity).

- **Lockdown**: The duration of the lockdown is fixed at 14 days, and it starts after the detection of the first infectious sample. The algorithm randomly choose 90% of the population for lockdown and compensate all of them according to *Daily quarantine compensation (compulsory isolation)* in Table 5.2. After the lockdown is lifted, the disease spreads according to the SIRD model and we expect the same peak as in DN but shifted over time. Note that this model adds a new state Q, for *Quarantined*, to Figure 5.1 with $Q_0 = 0.9|V|$, but does not change the equations.

- **TTI**:

  – *K-hop Ranking for* $k \in \{1, 2\}$: Prior to the simulation, each node is ranked according to the size of its k-hop neighborhood. For example, if the rank of a node is 16, it means this node, if infectious, can potentially contaminate 16 other individuals. In the tracing stage of TTI, it is chosen to forcibly isolate neighborhood of an infected node who have a ranking above 90 percentile of all rankings in the graph. The algorithm also chooses the neighborhood with ranking above 80 percentile (and less than 90 percentile) as candidates to voluntarily quarantine themselves. Both of these groups (forced quarantine and volunteer quarantine) are compensated but with different amounts (see *Daily quarantine compensation* for compulsory and volunteer isolation in Table 5.2). It is assumed the candidates for volunteer isolation accept the offer 50% of the times. Note that both of these thresholds can be chosen via trial and error and do not require global information on the graph.

  – *Random Ranking*: In the tracing stage of TTI, this method randomly choose candidates from the k-hop neighborhood ($k \in \{1, 2\}$) of an infected node to isolate. The isolated nodes are compensated as in lockdown.

- **CI**: This method obtains community memberships through Louvain partitioning [14]. At each timestamp, CI isolates an entire community if the portion of infected nodes within the community is greater than a threshold, i.e. $\frac{I_c}{|V_c|} > T$ where $|V_c|$ is the population within community c and T is a hyperparameter. I report the results for $T \in \{0.1, 0.2, ..., 0.9\}$. The isolated members are compensated as in lockdown.

Note that the choice of parameters such as disease duration and quarantine duration does not affect the comparison between Lockdown and TTI strategies as this parameter is set equally for all of them. In Section 5.5, I show that my comparisons are also robust against the choice of α, β, and γ.

5.4 Results & Discussion

The results of the DN, Lockdown, and TTI strategies are shown in Figure 5.2. In this figure, the y axis depicts the average (and variance) of the infectious population normalized by the overall population. In all datasets, the best performance is achieved through the k-hop neighborhood strategy. In 7 out of 10 datasets, the 2-hop strategy achieves a better performance and, in most cases, it is very close to that of 1-hop. However, looking at the required budget in Figure 5.3, it is evident that the choice of 2-hop neighborhood comes with a greater cost, especially for HX9 dataset.

To understand the reason behind enormous 2-hop budget for HX9, we need to look at the number of triangles in the network normalized by the maximum possible triangle count ($|V|(|V|-1)(|V|-2)$). Among all datasets, HX9 has the highest value for normalized triangle count (10 times more than the next highest count). Due to this high clusterability of HX9, the 2-hop neighborhood captures the entirety of the network and, in practice, gives a less optimal result than lockdown or even DN strategy in terms of budget. The special case with HX9 dataset shows the limitation of choosing 2-hop over 1-hop neighborhoods of the infectious nodes. Considering this trade-off between cost and peak of infection, we can conclude that 1-hop TTI strategy is the best practical strategy among the rest in real-world scenarios.

As the results for CI-based isolation are threshold-dependent, I have shown its results separately in Figure 5.4 and 5.5 for various threshold values. CI, surprisingly, does a much better job at reducing the infectious peak than any of the other methods (compare Figure 5.4 with Figure 5.2). This confirms the suggestions from psychology literature

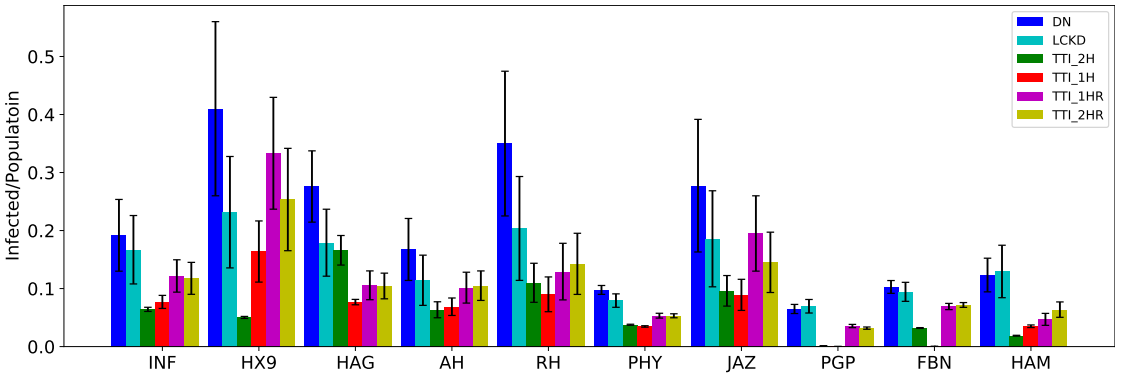


FIGURE 5.2: *The average proportion of infected individuals over 100 trials of simulation and its variance for each mitigation strategy among the chosen datasets. DN, LCKD, and TTI abbreviate Do Nothing, Lockdown, and Test-Trace-Isolate strategies. TTI suffixes: 1H and 2H represent k-hop ranking, 1HR and 2HR represent random ranking within k-hop neighborhood. Best viewed in color.*

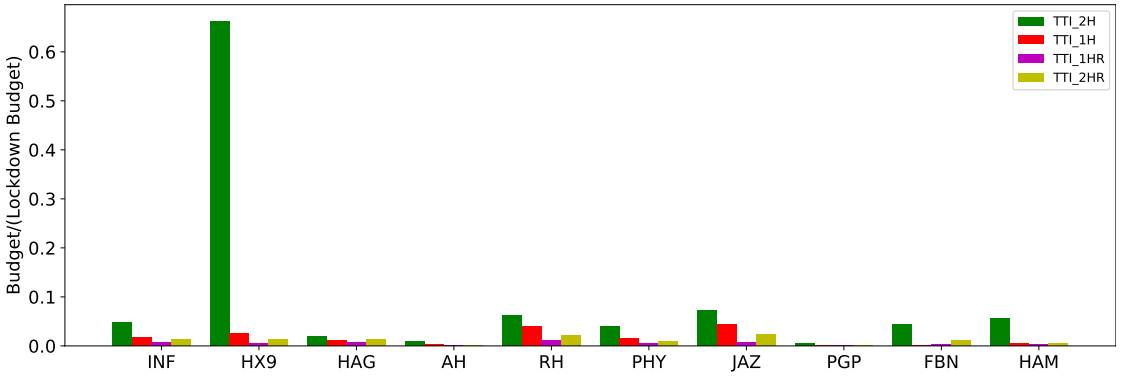


FIGURE 5.3: *The budget spent on isolation strategies. The budget is normalized by lockdown budget as the baseline. Best viewed in color.*

that mitigation strategies based on community membership can result in a better control over the speed of the spread.

The thresholding is very important for CI. As seen in Figure 5.5, for higher thresholds, CI generally comes with a much higher budget than TTI, and unlike the previous methods, surpasses the lockdown budget in many instances. However, keeping the threshold below 0.4 offers a considerable reduction in speed with a reasonable budget.

Note that the proposed CI strategy here uses limited information and still performs better than other strategies discussed in reducing the speed. CI strategy only requires the community assignment and the number of infected nodes within the community. This information is readily available through prior knowledge on the individual (e.g., the State or County of residence) and the information on the contagion progress (the [estimate] number of infected individuals). In practice, it is possible to gain more knowledge on

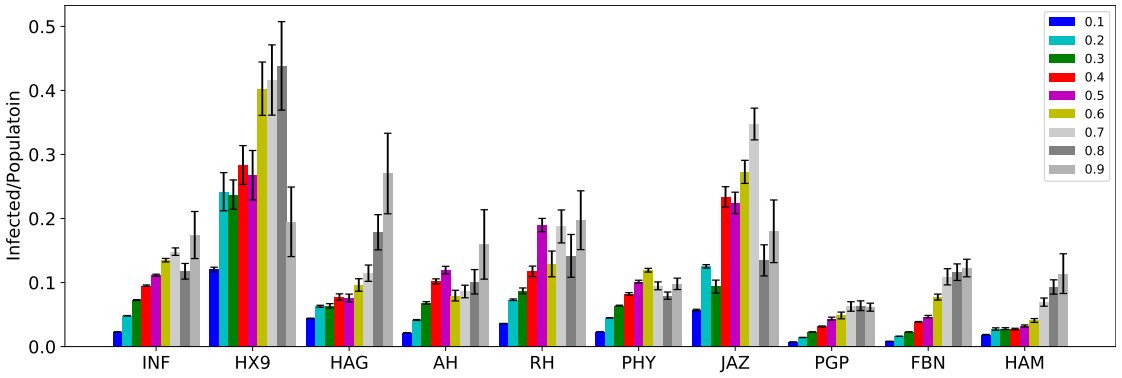


FIGURE 5.4: *The average proportion of infected individuals over 100 trials of simulation and its variance for different thresholds in CI (community-based isolation strategy). The lower thresholds give considerably better performance than strategies in Figure 5.2. Best viewed in color.*

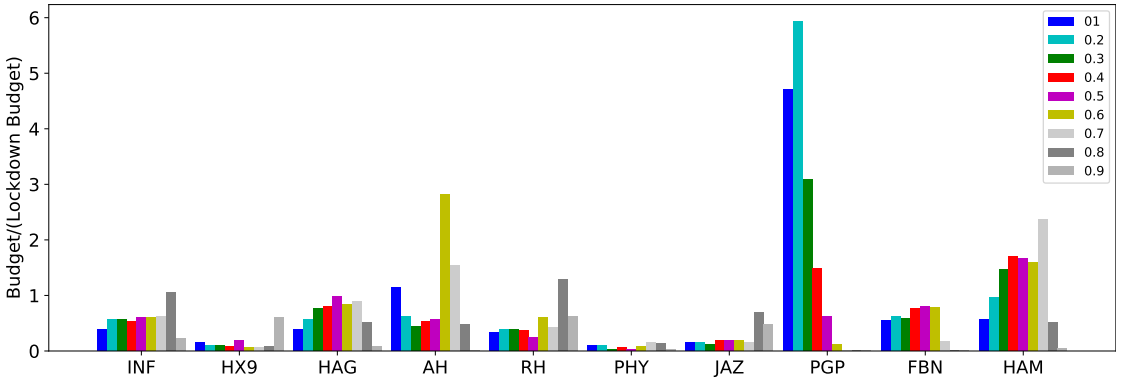


FIGURE 5.5: *The budget spent on CI for different thresholds. The budget is normalized by lockdown budget as the baseline. Except for PGP, the budget for lower thresholds among all datasets are comparable to those in Figure 5.3. Best viewed in color.*

the neighborhood of the infected nodes in the community (e.g., through contact-tracing and personal questionnaires upon testing) and use the neighborhood information for targeted isolation.

The special case of PGP. In Figure 5.5, all datasets follow the same trend that lower (higher) thresholds demand lower (higher) budget, except for PGP. A closer look at PGP community structure reveals large communities with high density (i.e., community structure that is considerably close to complete graph structure). In other words, the average path length between nodes in these communities is small and the virus spreads throughout the community much faster than in other datasets. As such, PGP meets the isolation threshold in CI much sooner than the rest of the datasets and forces communities of large size (i.e., containing many nodes) into quarantine. However, the higher speed of spread means higher number of deceased as well. As the CI threshold in

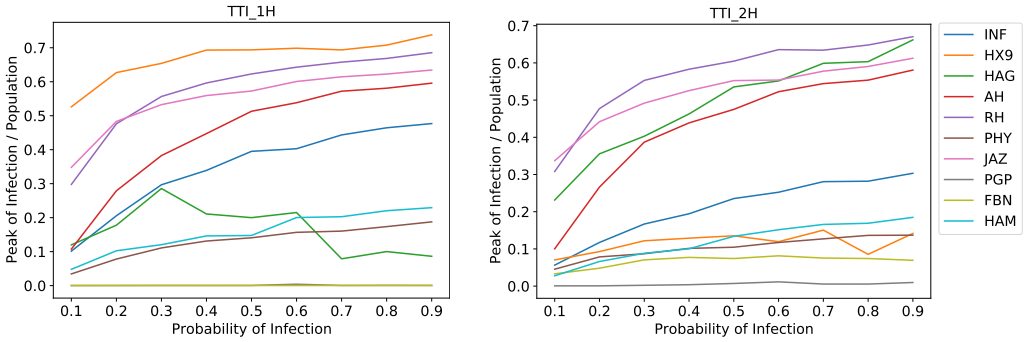
Figure 5.5 grows, it becomes harder for PGP to meet the isolation threshold as many of its members are dead and the CI threshold is defined over the primary size of the community. Hence, less and less communities are put into isolation (the decreased budget in Figure 5.5) and the peak of infection in Figure 5.4 matches that of DN in Figure 5.2.

This interesting example shows the limitation of CI in networks with communities that are close to complete graphs. For these type of communities, the 1-hop TTI gives the best trade-off between the peak of infection and budget. However, as is evident from the physical-contact datasets, the real-world human-human contacts have low-density communities and obtain better trade-off using CI strategy.

The choice of community. Throughout this study, I have defined community based on the structural property of the network (e.g., Louvain method). This definition of community expands to real-world communities of people within certain geographical region (e.g., state, county, city) that have more connection within the community than outside. However, there are also attribute-based communities that do not necessarily yield the same structural property. For example, a community defined based on gender, age, and race is not guaranteed to form communities that are dense inside and sparsely connected outwards. Just like communities in online social platforms such as Amazon that do not represent human contact for the modeling purposes of a viral spread, the attribute-based communities may not be a suitable candidate for my proposed community-based mitigation strategy, CI.

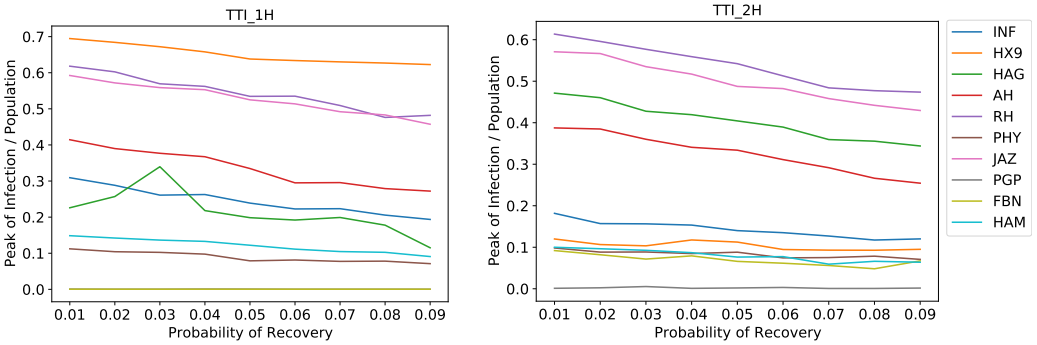
The superiority of CI in mitigating the spread shows that designing an optimal community-based strategy for further alleviation of the economic impact is a promising research direction. Moreover, community information can be local and noisy (through individual self-reported or publicly known memberships such as geographical proximity in a region).

My effort is to encourage more research on community-based mitigation strategies rather than brute-force methods such as lockdown or naïve TTI. Although k-hop and community-based methods seem to require extra effort for tracing the impact, they are still practical in real-world. My results show that with local approximation of network's structure, we still can obtain solutions that reduce both the physical and economic impact of the pandemic in a global scale.



(a) 1-hop Test-Trace-Isolate (b) 2-hop Test-Trace-Isolate

FIGURE 5.6: *Sensitivity of the peak of infection against probability of infection (α). The changes in the value of α affects the datasets in a mostly similar way and does not change the experiment result in Section 5.4. Best viewed in color.*

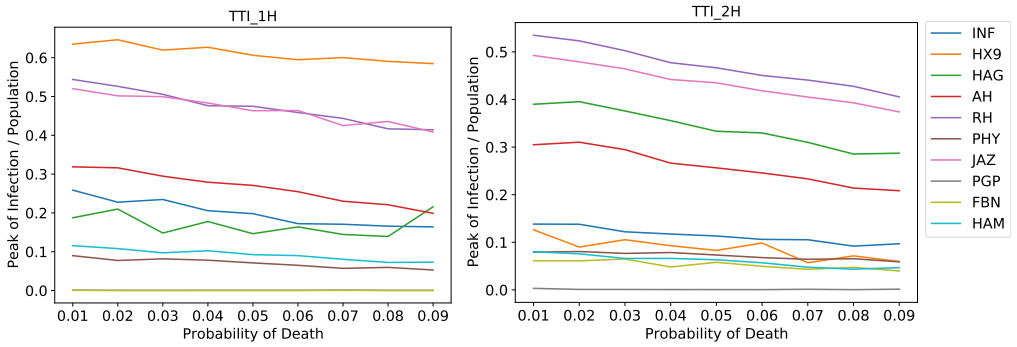


(a) 1-hop Test-Trace-Isolate (b) 2-hop Test-Trace-Isolate

FIGURE 5.7: *Sensitivity of peak of infection against probability of recovery (β). The changes in the value of β affects the datasets in a mostly similar way and does not change the experiment result in Section 5.4. Best viewed in color.*

5.5 Ablation Study

As mentioned in Section 5.3.3, my choice of hyperparameters in Table 5.2 does not change the result of the comparative study among different mitigation strategies. Here, I show the robustness of the results against the three degrees of freedom (α, β, and γ) in SIRD model. I repeat the same experiment in Figure 5.2 for different values of these parameters and report the results on peak of infection for 1-hop and 2-hop Test-Trace-Isolate in Figures 5.6 to 5.8. The results for other models were similar and are not included to avoid repetition. As evident from these three figures, the rate of change in peak of infection is mostly similar across all datasets and does not change the comparative observations in Section 5.4.



(a) 1-hop Test-Trace-Isolate

(b) 2-hop Test-Trace-Isolate

FIGURE 5.8: *Sensitivity of peak of infection against probability of death (γ). The changes in the value of γ affects the datasets in a mostly similar way and does not change the experiment result in Section 5.4. Best viewed in color.*

5.6 Summary

In this chapter, I analyzed the current state-of-the-art mitigation strategies that deal with early stages of a pandemic. The result of my analysis show that using the local neighborhood of nodes is enough to maintain a balance between reducing the coverage of the spread and minimizing the budget. I also showed that the mitigation strategy becomes even more powerful if we incorporate the community information, as expected from the results in Chapter 4. All these strategies, however, are in the ideal setting in which we possess the full knowledge of the social (contact) network. I use the results from the ideal setting in this chapter and, in next chapter, propose a dynamic mitigation strategy that overcomes this shortcoming.

Chapter 6

Blind Community-based Early Mitigation Strategy against Viral Spread

In response to a viral spread, multiple factors determine the efficacy of different mitigation strategies, namely the epidemiological knowledge of the spread dynamics, the possibility of medical intervention (i.e., vaccination), and the existence of mobility and interaction data [82, 151–153].

In the early stages of a pandemic, the disease dynamic is unknown, the contact network is partially known at best, and no vaccination is available. These are the challenges against an *Early Mitigation Strategy*. As we saw in the previous chapter, the objective of such a strategy is to minimize the magnitude of infection with the least possible perturbations introduced to the social network (through, e.g., quarantine and isolation approaches) [46, 127, 152]. In contrast, once a vaccination is available, the objective of the *Immunization* problem is to minimize the amount of time it takes to halt the spread effectively with the least amount of vaccine. Despite the similarity of approaches, immunization problem and early mitigation strategy optimize different objective functions and the former does not introduce perturbations to the network; the candidate nodes chosen in an early mitigation setting will be isolated for the duration of the disease (removal of edges in the contact network), but in immunization problem they are vaccinated and no network perturbation is introduced.

The majority of studies on the two mentioned strategies are based on detecting globally influential nodes (e.g., degree centrality [154] and betweenness centrality [28]) that

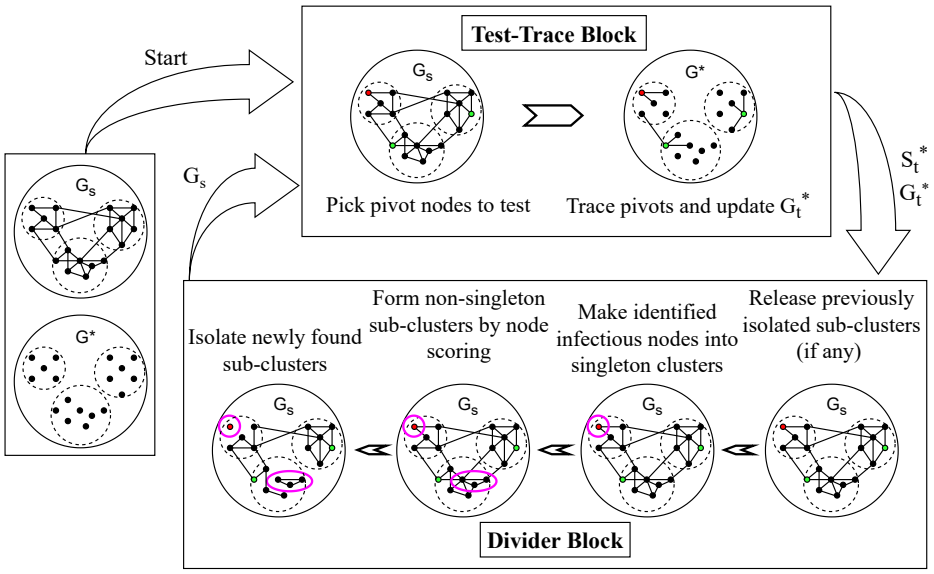


FIGURE 6.1: *ComMit pipeline.* ***Start.*** *Contact network G_s is unknown and the known graph to the algorithm, G^*, is empty. The dashed lines show the communities known to the algorithm. The figure shows the first iteration of the algorithm.* ***Test-Trace.*** *The coloring of the pivot indicates the result of the test (red is infectious). Tracing of the pivots updates the edges of G^*.* ***Divider.*** *The block uses the updated information on G_t^* and identified infections from S_t^* to form sub-clusters (in purple), whose isolation fragments the communities, reducing the magnitude of the spread. The network perturbations by divider updates G_s on which the spread runs. The iteration continues until the termination condition is met (see Section 6.3.3).*

contribute the most to the spread (*targeted strategies*). Despite promising theoretical results, these methods are generally difficult to implement due to their assumption of full knowledge on the contact network [155]. Additionally, complex social networks demonstrate high clustering and individuals tend to form groups (communities) [156], which can both alleviate and aggravate viral expansion [148, 151]. The global centrality measures do not consider the local influence of the node in their respective communities [152]. I argue that for tackling these shortcomings, a practical mitigation strategy, unlike the those in Chapter 5, should not assume a prior knowledge of the contact network structure and the dynamics of the spread. It also should consider the cost of a certain intervention scheme and avoid isolating healthy members of the population.

In this chapter, I study the problem of developing an early mitigation strategy from a *community perspective* and propose a dynamic Community-based Mitigation strategy, *ComMit*, that only utilizes geographical information to infer community membership and data from test-trace to update its knowledge of the spread, without enforcing any assumptions about the nature of the disease. Because *ComMit* relies on updated data from test-trace reports, it is dynamic and the mitigation strategy can evolve over time.

Unlike previous works, I have designed *ComMit* with two important assumptions in mind: (1) there is no global information on the social network contacts; (2) the candidates for isolation are small clusters instead of single healthy individuals. The second condition aims to minimize the economic and psychological damage ([157], [158]).

Using the information from the test-trace step, *ComMit* introduces appropriate network perturbations to combat the magnitude of the spread based on the current knowledge of the underlying network and testing outcome. These perturbations are aimed to fragment the network communities. *ComMit* achieves that through the *divider* block that forms small clusters of nodes (sub-clusters) that are to be temporarily isolated as a community from the rest of the network. After a certain time has passed these sub-clusters are released back to the network and will not be isolated until some time has passed from their last isolation. The pipeline for *ComMit* is shown in Figure 6.1.

My contributions in this chapter can be summarized as follows,

- I formulate the early mitigation problem based on real-world constraints.
- To the best of my knowledge, this is the first study proposing an early mitigation strategy that (1) works with no knowledge of either the social network structure and the spread dynamic; (2) considers the practical cost of the strategy and operates within a limited budget.
- I validate my mitigation strategy, *ComMit*, on five real-world datasets that are obtained from national address database and Copenhagen project (see Section 6.4).
- The result of my experiments show that within its limited budget, *ComMit* is very effective in reducing the peak and duration of infection, reducing them up to 73% and 90%, respectively. In all of the case studies, *ComMit* successfully turns a teady-state spread process[1], such as SIS contagion model, into a dying process with a relatively short absorption time.[2]

[1] A spread dynamic that reaches a steady state of maintaining a non-zero number of infectious nodes.
[2] The time that it takes for the number of infectious nodes to become zero.

6.1 Related Work

This chapter involves three bodies of research; targeted intervention strategies against viral spread, the impact of community structure on dynamics of such a spread, and community-based intervention strategies.

6.1.1 Targeted Intervention Strategies

Early Mitigation Strategy

Gross and Havlin [62] perform the closest study to ours in modeling a contact network based on geo-spatial data. Their contact network model is a modular 2D lattice in which each module represents a city and each city can only connect to its immediate neighboring module. Their proposed mitigation strategies are social distancing and reducing degrees in and outside of the communities by isolating individuals. My approach differs in that *ComMit* (1) does not limit contact network to a 2D lattice; (2) considers a mixture of sampling (testing) and isolation; (3) does not isolate any healthy individuals; (4) does not assume complete knowledge of the contact data.

Immunization Strategy

Rosenblat et al. [82] challenge the popular target-based immunization strategies by studying different immunization methods in the presence of partially observed network data. They conclude that popular targeted methods, such as degree and betweenness centralities, only perform well with little to no missing data, but self-reported local information from sampled individuals compensates for a large volume of missing data.

Salathé and Jones [25] place a similar emphasis on partially observed networks and propose a heuristic method to find community bridge nodes (CBF that stands for Community Bridge Finder). They show how targeting bridge nodes for immunization outperforms acquaintance immunization (the only other network structure agnostic method) [159], in which the most frequently visited acquaintances of randomly selected nodes are vaccinated first. CBF relies on random walks and constant path finding between current and visited nodes. This limits its value in a practical dynamic setting where these computations need to run iteratively. In my method, I show how we can avoid

such costly (and impractical) computations by leveraging the known geo-spatial communities that are shown to be predictors of contact-based communities [160, 161].

6.1.2 Community Structure and Dynamics of Spread

Topîrceano [149] shows the importance of geo-spatial information in predicting the dynamics of an outbreak. This paper offers a Geo-spatial Population Model (GPM) that estimates the predictors of mobility between different regions in a country based on the region's population density. His results suggest that changing the number of regions and their population density directly impacts the size and duration of the outbreak.

6.1.3 Community-based Intervention Strategies

The definition of community in these studies is diverse: from subgraphs with the highest $\frac{\text{number of subgraph intra edges}}{\text{number of subgraph inter edges}}$ and k-cores to geo-spatial and ground truth communities. Serafino et al. [46] showed that disconnecting bridge nodes that connect super-spreader k-cores considerably reduces the radius of the spread. However, they rely on betweenness centrality which is a global measure that requires full knowledge on contact network.

Yang et al. [162] propose a flow-based edge betweenness measure to minimize the p-norm of the flow between communities in the network. They show that the bridge-based methods are superior to degree-based intervention methods.

Block et al. [157] consider social behavior patterns within communities and propose to (1) limit interaction to few repeated contacts; (2) choose those contacts based on some similarity (e.g., homophily); (3) strengthen contact with those pairs that interact in more than one community. I leverage their finding and that of Topîrceano [149] to design the *fragmentation* step in *ComMit* (see 6.3.2). The main problem with their strategy is the assumption of full network knowledge.

Yuan and Tang [152] emphasize the local importance of the nodes to their community in contrast with their global centrality. They measure how important nodes are to their communities and how important their communities are to the overall network. Their scores are based on eigenvalue and eigenvector pairs obtained from the spectral clustering of the neighborhood matrix. This method is susceptible to edge percolation and loses its performance with partial network data.

Symbol	Definition
Input to blind network fragmentation problem	
$G^*_t = (V, E*_t)$	Learned contact social network at time t
$S^*_t(.)$	Partially known outcome of $S_t(.)$ at time t
$\mathcal{I}^*_t$	Set of identified infected nodes at time t
b_f	Budget for forming sub-clusters (divider block)
Additional input to ComMit algorithm	
$G_g = (V, E_g, W_g)$	Geo-proximity network
C_g	Geo communities inferred from G_g
b_t	Budget for testing (test-trace block)
a_t	Accuracy of self-reports in contact tracing
ϵ_t	Value of ϵ for testing at time t
d_ϵ	Decaying factor for updating ϵ_t
t_r	Restriction period of isolating sub-clusters
Other notations	
$\mathfrak{C}_t$	Set of sub-clusters at time t
$G_s = (V, E_s)$	Contact social network
$S_t(.) = \{s^v_t \mid v \in V\}$	Outcome of spread at time t
$\mathcal{I}_t = \{v \in V \mid s^v_t = I\}$	Set of infected nodes at time t

Table 6.1: *Notations.*

6.2 Problem Statement

Inspired by the findings of Block et al. [157] and Topîrceano [149], I argue that fragmenting network communities into small clusters (*sub-clusters*) and isolating these sub-clusters, rather than isolating individuals is the best strategy during the early stages of a contagion. I refer to the problem of finding such sub-clusters as the *Network Fragmentation Problem* and it is the backbone of the *ComMit* algorithm.

Assume the beginning of an unknown viral infection within an unknown contact network. with a known underlying geo-spatial structure (e.g., the geographic coordinates of domiciles). Consider that we have the power of restraining individuals to limit their interactions within a certain group in exchange for a compensation. This introduces perturbations in the underlying unknown contact network that changes the dynamic of the spread. The main question is how to choose groups of individuals such that isolating them as a group from the rest of the network, while maintaining their inner-group interaction, most efficiently inhibits the spread. This is the network fragmentation problem that I formally define in Section 6.2.4, but first, I discuss the population model, contagion model, and assumptions on network perturbations , as follows. The notations used in this and next section are summarized in Table 6.1.

6.2.1 Population Model

Empirical studies on human contact have shown geo-spatial distance to be the most important factor in forming connections [160]. More recent studies on online social networks show the geo-spatial distance also influence the presence of online contacts and they are inversely correlated by a power-law [161]. This observation can be used to compensate for an absence of knowledge on contact network structure.

I model the population as a two-layer network consisting of the contact network $G_s = (V, E_s)$ and its underlying geo-location graph $G_g = (V, E_g, W_g)$. Both layers are undirected and share the same set of nodes, V. G_g is a complete graph and a weighted edge $(i, j, {w_g}^{ij}) \in E_g$ indicates a geo-distance of ${w_g}^{ij}$ between nodes i and j. G_s, however, is sparse and an edge $(i, j) \in E_s$ implies the existence of contact between nodes i and j. I assume the distance between individual domiciles and their contact patterns do not change.

The community membership of each node is inferred from G_g, while the infection spreads through the links in G_s. The key underlying assumption is the inverse relationship between W_g and E_s, as demonstrated in [160, 161]. More specifically, the empirical results in [161] suggest a Zipf's law:[3] i.e., the probability of an edge between nodes (i, j) in G_s is

$$p((i, j) \in E_s) \approx \frac{b}{w_g^{ij}}, \quad 0 < b \leq 1, \quad 1 \leq w_g^{ij} \tag{6.1}$$

for a constant b. I use this rule in building the datasets in Section 6.4. From the perspective of a mitigation strategy, E_s is partially known. I represent this partially known network at time t by $G_t^* = (V, E_t^*)$. In each iteration G_t^* is updated by the information from test-trace (Figure 6.1). If nothing about E_s is known (i.e., in the start of the algorithm, or in the absence of test-trace block), E_t^* is empty.

6.2.2 Contagion Model

Consider a viral spread with unknown dynamics, $S_t(G_s) = \{s_t^v | v \in V\}$, that impacts the contact network G_s by changing the state of nodes in V at each timestamp. In this definition, s_t^v denotes the state of node $v \in V$ at time t, and $S_t(.)$ is a graph function

[3] In [161], the exponent of the best power-law fit if sound to be -1.03 with a standard error of 0.03, which can be approximated by a Zipf's law.

whose domain and range are $|V|$ and a pre-defined set of possible states, respectively. The only known facts about $S_t(G_s)$ from the perspective of an early mitigation strategy are (1) infectious ($s_t^v = I$) is one of the possible states, and (2) the infection spreads through direct contact.

6.2.3 Network Perturbations

The only network perturbations required for the network fragmentation problem are edge deletion and edge addition. The edge addition is only limited to the edges that have been previously deleted by the algorithm (isolation process) and are to be released. Since one of the criteria for the early mitigation strategy is to minimize the isolation of healthy individuals, the selection of edges for perturbation is performed through selection of sub-clusters of nodes. The healthy individuals are restricted through isolation of these sub-cluster; i.e., the members of a sub-cluster can only contact others within the sub-cluster and not outside of it. This means the inter-cluster edges of the sub-cluster will be preserved while the intra-cluster edges are removed.

To limit the amount of network disturbance (e.g., due to economic cost), there is a budget for the selection of sets of nodes to form sub-clusters. This budget, which I refer to as b_f, represents the cost of restricting the movement of individuals in a network (e.g., daily monetary compensation). As such, it is logical to consider b_f in terms of the number of restricted nodes per timestamp rather than the number of edges that are perturbed (e.g., we pay restricted individuals the same compensation regardless of their number of contacts).

6.2.4 Network Fragmentation Problem Statement

Given the contact network $G_s(V, E_s)$, the outcome of a temporal spreading process $S_t(.)$, and a fragmentation budget b_f, the network fragmentation problem is to find a set of sub-clusters $\mathfrak{C}_t(G_s, b_f)$ at time t whose isolation minimizes the total number of infectious nodes at time $t + 1$. In formation of these sub-clusters, only *known* infectious nodes are allowed to form singleton sub-clusters. Formally,

$$\begin{aligned}
\mathfrak{C}_t(G_s, b_f) = \min_{G_s, S_t(.)} \; & |\mathcal{I}_{t+1}| \\
\text{s.t} \quad & \sum_{C \in \mathfrak{C}_t(G_s, b_f)} |C| \leq b_f \\
& |C| > 1 \;, \forall C \in \mathfrak{C}_t(G_s, b_f) \quad \text{if} \quad s_t^v \neq I \;, \forall v \in C \\
& |C| = 1 \;, \forall C \in \mathfrak{C}_t(G_s, b_f) \quad \text{if} \quad s_t^v = I \;, \forall v \in C.
\end{aligned} \tag{6.2}$$

With known G_s and $S_t(.)$, the answer to this problem is trivial: putting all infectious nodes in $\mathcal{I}_t$ in singleton sub-clusters and isolating them gives the optimal solution.

The problem is non-trivial once we add the assumptions of the early mitigation strategy: partially known G_s and $S_t(G_s)$ at time t, which are shown as G_t^* and S_t^* in Table 6.1, respectively. This problem, which I will refer to as *Blind Network Fragmentation Problem*, is then formulated as follows,

$$\begin{aligned}
\mathfrak{C}_t(G_s, b_f) = \min_{G_t^*, S_t^*(G_s)} \; & |\mathcal{I}_{t+1}| \\
\text{s.t} \quad & \sum_{C \in \mathfrak{C}_t(G_s, b_f)} |C| \leq b_f \\
& |C| > 1 \;, \forall C \in \mathfrak{C}_t(G_s, b_f) \quad \text{if} \quad s_t^{*v} \neq I \;, \forall v \in C \\
& |C| = 1 \;, \forall C \in \mathfrak{C}_t(G_s, b_f) \quad \text{if} \quad s_t^{*v} = I \;, \forall v \in C,
\end{aligned} \tag{6.3}$$

in which $s_t^{*v} \in S_t^*$. Note that the difference between 6.2 and 6.3 is that 6.3 uses the information from partial observations, G_t^* and S_t^*, to minimize $\mathcal{I}_{t+1}$. *ComMit* is a heuristic algorithm that aims to minimize 6.3. The next section outlines its details.

6.3 Method

Here, I introduce the *ComMit* algorithm for dynamically perturbing a network to inhibit the progress of a viral spread, as defined in 6.3. *ComMit* does not require a priori knowledge of the contact network structure. Other methods with a similar assumption (which mainly deal with immunizations) [25, 82, 159], overcome this limitation by relying on extensive sampling from the contact network (in the form of random walks

and/or random node sampling). In practice, assuming there is an unlimited budget for sampling is unrealistic.

Another assumption of *ComMit* is blindness to the dynamic of the spread, which in turn calls for an efficient testing strategy to identify as many infectious nodes as possible. Although the intuition behind sampling and testing is different (one tries to learn about the network structure whereas the other aims to locate the infectious nodes), the mechanism by which they operate is the same: they select candidates from the pool of nodes in the network based on certain criteria and both within a limited budget in real-world scenarios. Considering their similarity, I combine the sampling and testing into one temporal algorithm. At each timestamp, the goal of this algorithm is to update *ComMit*'s knowledge about the network structure and the infectious hubs simultaneously. I refer to this algorithm in the *ComMit*'s pipeline as *test-trace block*. Iteratively, the output of this block is fed into the *divider block* in which the fragmentation-based mitigation strategy of *ComMit* perturbs the network to inhibit the spread (Figure 6.1). Below, I discus the details of these two blocks.

6.3.1 Test-Trace Block

As evident from the name, the test-trace block consists of two steps: **Testing.** The selection of candidates (pivots) from the population to be tested. This step determines whether these candidates are infectious or not. **Tracing.** Contact tracing of pivots in order to update the known contact network, G^*. Note that the traced contacts will not be tested.

Consider a temporal **testing strategy**, $T_t(G_s, b_t) = \{s_t^v | s_t^v \in S_t(G_s)\}$ with limited budget b_t, whose purpose is two-fold: (1) finding as many infectious nodes in $\mathcal{I}_t$ as possible; (2) gathering information about unknown G_s network to update the known G_t^* network. More formally, an optimal testing strategy would minimize the following,

$$T_t(G_s, b_t) = \min_{G_s, S_t(G_s)} \text{dist}(G_s, G_t^*) + \text{dist}(S_t(G_s), S_t^*(G_s))$$
$$\text{s.t.} \quad |T_t(G_s)| \leq b_t, \tag{6.4}$$

in which $dist(a, b)$ denotes the distance between a and b. This problem is similar to the exploration-exploitation scenario.

A well-known algorithm to address the exploration-exploitation problem in machine learning is ϵ-*greedy*. This algorithm selects an action from a set of possible actions based on a given reward function; the action that maximizes the reward function is selected with probability $1-\epsilon$ and a random action is chosen with probability ϵ. I adapt this idea to the graph-based exploration-exploitation problem and select the pivots as follows,

$$\text{pivot}_t = \begin{cases} \text{randomly choose from } \mathcal{I}^*_{t-1}, & p(1-\epsilon_t) \\ \text{randomly choose from } V, & p(\epsilon_t) \end{cases} \tag{6.5}$$

in which $\mathcal{I}^*_{t-1}$ is the set of infected nodes identified in the previous timestamp. At each time, *ComMit* selects as many pivot nodes as allowed by b_t. As time progresses, it has more knowledge about the network and can rely on exploitation more than exploration. To make that possible, the value of ϵ is updated through a decaying factor d_ϵ as,

$$\epsilon_t = \max\left(\epsilon_{t-1} - \frac{\epsilon_{t-1}}{d_\epsilon}, 0\right), \quad d_\epsilon > 0. \tag{6.6}$$

Once the pivots are tested and $\mathcal{I}^*_t$ is updated, the **tracing strategy** is straightforward: the pivots are asked to provide the information about their immediate neighborhood. This information may have less than 100% accuracy. I denote this accuracy by a_t and study its impact in Section 6.5.2. The new edges obtained from tracing update G^*_t which will be used by the divider block.

6.3.2 Divider Block

The divider is the main building block of *ComMit* that handles the network perturbations aimed at decreasing the magnitude of the spread. The intuition behind divider is to fragment the bigger communities by reducing the density of its inter-connections. Using the updated $\mathcal{I}^*_t$ and G^*_t from test-trace, the divider identifies a new set of sub-clusters, $\mathfrak{C}_t$, to be temporarily isolated from the network. It does so by attributing a score to each candidate node for forming a sub-cluster. The score is calculated for the community $C \in C_g$ of each node, where C_g is the geo-communities inferred from G_g.The scoring function has three components:

$$\text{score} = \frac{1}{3}(\text{norm-size} + \text{inf-rate} + \text{density}), \tag{6.7}$$

$$\text{norm-size} = \frac{|C|}{|V|}, \tag{6.8}$$

$$\text{inf-rate} = \frac{|\{v \in C | s_t^{*v} = I\}|}{|C|}, \tag{6.9}$$

$$\text{density} = \frac{1}{|E_t^*|}(|\{(v_1, v_2) \in E_t^* | v_1, v_2 \in C\}| - |\{(v_1, v_2) \in E_t^* | v_1 \in C, v_2 \notin C\}|), \tag{6.10}$$

which, in order, are: (1) normalized community size, (2) proportion of nodes within community that are known to be infectious, and (3) community density as the proportion of edges in the known contact network that are inside of the community (i.e., excluding the outgoing edges). The nodes within a community all have the same score. The divider randomly picks b_{fs} candidates from the top 20^{th} percentile of the scores as the seed for the sub-cluster. It ensures the sub-cluster is not singleton by randomly adding b_{fn} neighbors of each seed that are available (e.g., not isolated with another sub-cluster) to the sub-cluster. Hence, the overall budget of the divider is $b_f = b_{fs} \times b_{fn}$. The isolation of a sub-cluster refers to cutting all the outgoing edges from a sub-cluster while maintaining the edges inside.

The divider is also responsible for releasing the currently isolated sub-clusters that have served their isolation time (t_r). To assure these released sub-clusters do not get restricted again and indefinitely, divider places the members of these sub-clusters in a *banned list* that inhibits these nodes nodes from forming another isolated sub-cluster for at least t_r time. The steps for the divider algorithm at time t are,

1. Release sub-clusters isolated at time $t - t_r$.
2. Add the members of the released sub-clusters in the banned list and remove those who have been in the list for t_r time.
3. Put recently identified infectious nodes ($\mathcal{I}_{t-1}^*$) into singleton sub-clusters.
4. Calculate the community score according to 6.7 for nodes that are neither in an isolated sub-cluster nor in the banned list.

5. Pick b_{fs} seed nodes with the score in the top 20$^{\text{th}}$ percentile of scores and form their sub-clusters by selecting b_{fn} of their neighbors at random. If the neighboring list is empty, remove the corresponding seed node from the candidates.

6. Remove outgoing edges of the new sub-clusters to isolate them.

6.3.3 ComMit Algorithm

Combining the *test-trace* and *divider* blocks into a pipeline that iteratively perturbs the contact network yields the final *ComMit* algorithm (see Algorithm 6). An important note is when ComMit terminates. Ideally, it would terminate when either there is no more budget allocated from time t onward, or the spread has dies out (i.e., $|\mathcal{I}_t| = 0$). Since $\mathcal{I}_t$ is unknown, I set the latter **termination condition** such that if for T consecutive timestamps no new infectious node is found (i.e. $|\mathcal{I}^*_{t_i}| = 0, \quad \text{for} \quad t_i \in \{t-T, t-T+1, ..., t\}$), the spread is considered eradicated.

Algorithm 6: `ComMit()`

Input: $V, G_g, t_r, b_{fn}, b_{fs}, b_t, \epsilon, d_\epsilon$

$C_g \leftarrow$ `ExtractCommunities`(G_g)

$S^*_0 \leftarrow \{s^{*v}_0 = \bar{I} | v \in V\}$ // $\bar{I}$ = non-infectious

$G^*_0 \leftarrow (V, \{\})$; $\quad \mathcal{I}^*_0 \leftarrow \{\}$; $\quad t \leftarrow 1$

/* iterate while termination condition not met */

while *NotTerminated()* **do**

  /* TestTrace() operates on unknown network, G_s */

  $\mathcal{I}^*_t, S^*_t, G^*_t \leftarrow$ `TestTrace`$(\mathcal{I}^*_{t-1}, S^*_{t-1}, G^*_{t-1}, \epsilon, d_\epsilon, b_t)$

  /* Divider() updates G_s on which spread runs */

  `Divider`$(\mathcal{I}^*_t, S^*_t, G^*_t, t_r, b_{fn}, b_{fs})$

  $t \leftarrow t + 1$

end

6.3.4 Budget Analysis

ComMit has two budgets: the testing budget (b_t as the number of nodes tested at time t) and the fragmentation budget (b_f as the number of non-infectious nodes that are members of restricted sub-clusters t). The latter is divided into two separate budgets; one for choosing the sub-cluster seed nodes (b_{fs}) and the other for selecting a certain number of known immediate neighbors of each seed node (b_{fn}). Empirically, we have witnessed that for b_{fn} values greater than 2 no significant performance gain is achieved

The other two budgets are expressed as a proportion of $|V|$: $b_{fs} = a|V|$ and $b_t = b|V|$. Thus, the total budget for *ComMit* becomes $(b_{fn} \times a + b)|V|$, which can be tuned by setting a and b accordingly (see 6.5.2 and 6.5.3).

6.4 Experimental Setup

In this section, I elaborate on the simulation setup, demonstrate the performance of *ComMit* on real-world datasets, and perform ablation studies to gain more insight about the strengths and limitations of the algorithm.

6.4.1 Contagion Model for Simulation

The majority of previous studies use SIR model in their simulations as the permanent immunity condition facilitates the analytical tractability [155]. To explore a less investigated direction, I consider the SIS model as the underlying dynamic of the spread. The SIS contagion model models viruses such as common cold, influenza, and COVID-19 [163]. In the SIS model, each node at time stamp t can either be susceptible (S) or infectious (I). The transition from S to I is controlled by the infection rate α. The infected nodes transition back to S once they pass the disease duration t_d. The default values for α and t_d in my experiments are 0.5 and 3, unless otherwise is specified. I initialize an infection by selecting $0.01 \times |V|$ nodes form the population uniformly at random. In all of the simulations, I use 10 different sets of initial infectious nodes (referred to as source nodes from here on).

I also report the performance of *ComMit* and other benchmarks on SIR model (similar to the model discussed in 5.1.2) in Section 6.5.1 and see that the true benefit of my proposed approach is in controlling the non-zero steady state contagions (i.e., large contagion magnitude) such as SIS.

6.4.2 Dataset

The ideal real-world dataset for testing my geo-social network model should contain the information on both the geo-locations and social interactions between the nodes. To the best of my knowledge, due to privacy concerns, such datasets are not available.

	Albany	Syracuse	Rochester	Copenhagen	Ithaca
$\|V\|$	2,858	2,385	1,312	512	127
$\|E_s\|$	4,641	1,756	4,742	1,416	315
$\|C_g\|$	4	4	5	16	3

TABLE 6.2: *Datasets general information (G_s).*

To navigate this problem, I use equation 6.1 and consider two types of data: *(A)* data with real-world geo-locations and their pairwise distance, and *(B)* the data with real-world social interactions and their pairwise probability of contact. I use the following strategies to process each category.

- **Constructing social network from geo network:** First map the pairwise distances to $[1, \text{inf})$ interval. Then, using the equation 6.1 with $b = 1$, obtain the probability of contact between each pair. Keep the edges with non-zero probability values (rounded to one decimal). Community membership is obtained via k-means clustering [164] with optimal k that minimizes the inertia.

- **Constructing geo network from social network:** First form the social network from mobility data with edge weights (w_s^{ij}) in $(0, 1]$. 6.1 for $b = 1$ gives the Geo network weights w_g^{ij}. This is a partially constructed geo network as some edges in the social network are non-existent. To complete the geo network, use the Weighted Shortest-Path (WSP) length between two nodes in the partially constructed geo network. (i.e., $\text{WSP}_{G_g}(.)$). So,

$$W_g = \begin{cases} \frac{1}{w_s^{ij}}, & (i,j) \in E_s \\ \text{WSP}_{G_g}(i,j), & \text{otherwise} \end{cases} \tag{6.11}$$

  The community memberships are obtained using Louvain algorithm [14] on the constructed geo-network.

NAD Dataset. For the first datatype, I use the U.S National Address Database (NAD)[4] (see 6.2) and build four different geo-networks: Syracuse, Albany, Rochester, and Ithaca. pairwise distance is computed using latitude and longitude.

Copenhagen Dataset. For the second datatype, I consider the Copenhagen Network Study Interaction Data [165] (see 6.2). In this study, students were followed through

[4] https://www.transportation.gov/gis/national-address-database/national-address-database-nad-disclaimer

their Bluetooth devices across the campus for 28 days. Every five minutes, the Bluetooth devices detected in their vicinity is recorded. Following the definition of close contact by CDC[5], I translate these recording as close contact if at least 15 minutes of contact is observed within a 24-hour interval for each pair of students. The social network weights are defined as average daily frequency of each close contact and are mapped into $(0, 1]$ interval to represent the pairwise probability of contact.

6.4.3 Evaluation Metric

Contagion Metric. A spread can be described by its (1) absorption time (the time it takes until no infectious node exists in the population, i.e., $|\mathcal{I}_t| = 0$); (2) the peak of infection. In the absences of vaccination, SIS spreads often reach a steady state of maintaining non-zero infection rather than absorption state. Hence, I limit the simulation time to 200 steps. I will show that *ComMit* effectively absorb the steady-state SIS infection in a short time for all datasets. In Section 6.5, I use these two metrics to compare different mitigation strategies.

Test-Trace Metric. The test-trace block in *ComMit* is designed to simultaneously find the most number of infectious nodes and edges of the social network through tracing. Hence, for comparing testing strategies, I consider three metrics,

- **Test Efficacy:** Defined as the ratio between number of infectious nodes found and total number of infectious nodes at each timestamp ($\frac{|\mathcal{I}_t^*|}{|\mathcal{I}_t|}$)
- **Test Efficiency:** Defined as the ratio between number of infectious nodes found and total number of tests taken at each timestamp ($\frac{|\mathcal{I}_t^*|}{b_t}$). This is the same measure as the *Positive Detection Rate* used in previous studies on efficient test strategies [166].
- **Kullback–Leibler Divergence:** The KL divergence between the degree distribution of unknown G_s and that of reconstructed G_t^* at each timestamp. The KL divergence between two probability distributions P and Q is defined in 3.17. In our case, P is the probability mass function of G_t^*'s degree distribution and Q is that of G_s.

[5] https://www.cdc.gov/coronavirus/2019-ncov/php/contact-tracing/contact-tracing-plan/appendix.html

In Section 6.5.2, I compare several test strategies based on these evaluation metrics.

Divider Metric. The main bottleneck of the divider block is the number of restricted nodes it requires to effectively improve the contagion metrics. I define this number as the divider metric (b_f) and evaluate the goodness of divider as the trade-off between this budget and the resulting contagion metric after implementing *ComMit*. This metric is used in Section 6.5 for benchmarking and ablation studies.

6.4.4 Benchmarks

To the best of my knowledge, there are no temporal mitigation strategies that consider all the limitations of the early-stages in a viral spread (ignorance of the network structure and dynamics of the spread, and limitation of the sampling and network perturbation budgets). For a fair comparison, I build the benchmarks by using the same test-trace method as in *ComMit* to give the advantage of efficiently probing the network within a limited budget. My benchmarks for the divider block of *ComMit* are as follows,

- **ComMit with CScore.** The original *ComMit* pipeline discussed in Section 6.3.
- **ComMit with DScore.** Similar to *ComMit with CScore*, but uses the degree centrality in G_t^* to score and choose seed nodes.
- **ComMit with IScore.** Inspired by test-based strategies whose goal is to find the most number of infectious nodes to isolate, the divider is changed such that it selects the seed nodes from the known infectious by their degree centrality in G_t^*.
- **Acquaintance immunization** Similar to acquaintance immunization method in [159], seed nodes and their neighbors are selected at random to form sub-clusters. Note that in this method there is no singleton sub-clusters and identified infectious nodes may or may not be included in the sub-clusters.
- **Community isolation.** Considering the good performance of community-based isolation (with known contact network) in the previous chapter, I use the information from test-trace to decide whether to isolate the entirety of a community. This method does not form sub-clusters. Once the ratio of the infectious nodes within the community surpasses a certain threshold, the community is isolated for the duration of t_r (the same value across all baselines). My experiments showed a threshold of 0.1 gives the best reasonable trade-off between the budget and performance.

- **No intervention.** The baseline without any mitigation strategy.

6.5 Results & Ablation Studies

In this section, I first report the comparative analysis of *ComMit*'s performance on inhibiting the infection in terms of the duration and magnitude of the spread. Next, I perform ablation studies on test-trace and divider block, respectively, to back up my hypotheses and methodology.

6.5.1 Inhibiting Contagion

SIS Contagion Model. The results of my simulations for SIS contagion model are shown in Table 6.3. In addition to the evaluation metrics, I also report the maximum divider budget for each strategy (the test budget is the same for all). The default values for hyperparameters are: $a_t = 1$, $b_t = 0.1 \times |V|$, $b_{fs} = 0.01$, $b_{fn} = 2$, and $t_r = 3$. The results show that *ComMit* variants, *ComMit with CScore* and *ComMit with DScore*, yield similar performance with the exception that the former has a shorter absorption time on average. The other two strategies, *ComMit with IScore* and *Community Isolation*, do not have guaranteed performance as in some cases they either do not terminate the spread or use an unrealistically large budget. *Acquaintance Immunization* consistently yields a poor performance across all datasets. At its best, *ComMit* reduces the peak of infection by 73% and the absorption time by 80% (see the first row for Albany). At its worst, it reduces the peak by 6% and the absorption time by 90% (see the first row for Rochester); a trade-off that still beats the other baselines.

Figure 6.2 is an example of changing spread dynamic for each strategy for Copenhagen dataset. Figure 6.5 shows the magnitude of restriction imposed by each strategy for the same dataset. They show that the community-based and degree-based *ComMit* resulting in the best trade-off between lowering the peak of infection, shortening the absorption time, and limiting the restriction magnitude.

SIR Contagion Model. Similar results are shown in Table 6.4 and Figures 6.4 and 6.5 for SIR contagion model. In general, the SIR model reaches its absorption time quickly in all datasets. I chose infection rate α to be 0.1 in order to prolong the duration of infection (although the absorption time is still small). Other parameters are the same as SIS model.

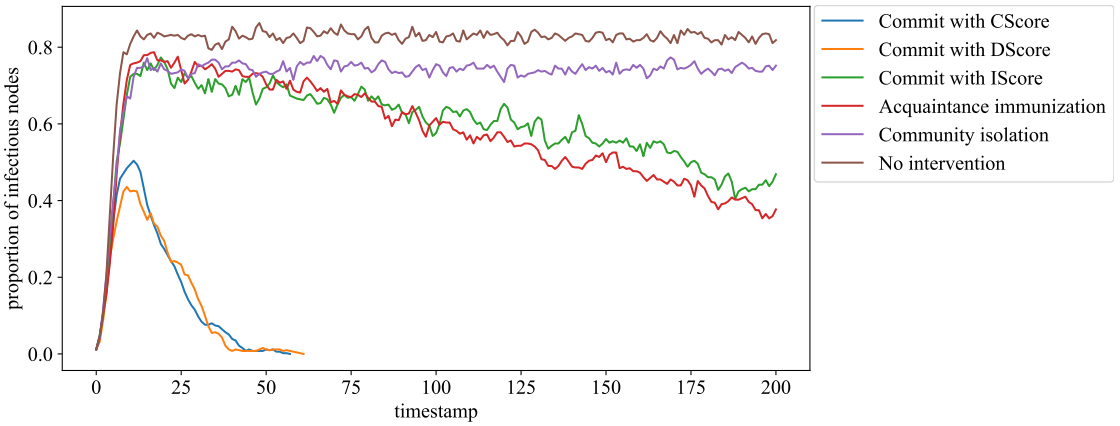


FIGURE 6.2: *The change in the dynamic of the spread due to mitigation strategies for Copenhagen dataset for SIS contagion model. The community-based and degree-based ComMit has the best performance in terms of lowering the peak of infection and shortening the absorption time.*

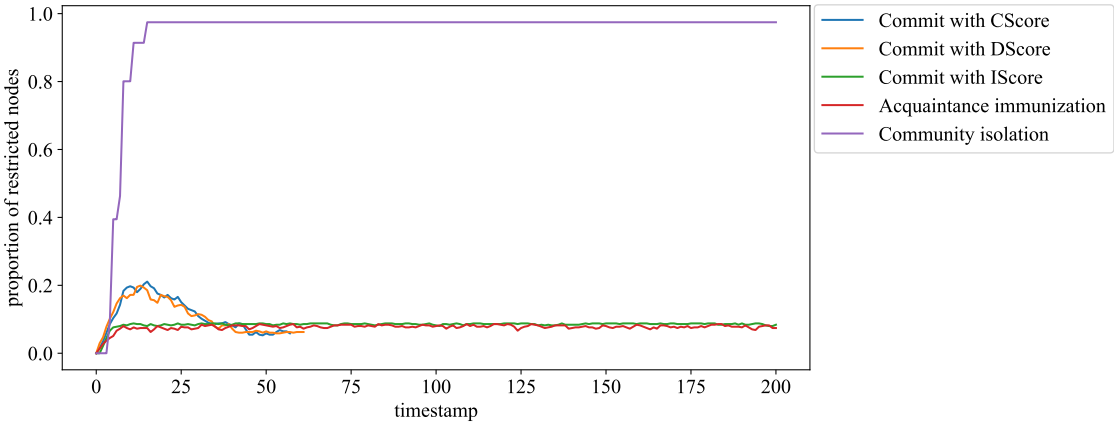


FIGURE 6.3: *The number of restricted nodes at each timestamp for Copenhagen dataset under SIS contagion model. The community isolation requires almost the full graph to be restricted. Community-based and degree-based ComMit restrict much smaller proportion of the population. Their magnitude of restriction is comparable to that of acquaintance immunization and commit with IScore, whereas the latter two cannot inhibit the SIS contagion as shown in Figure 6.2.*

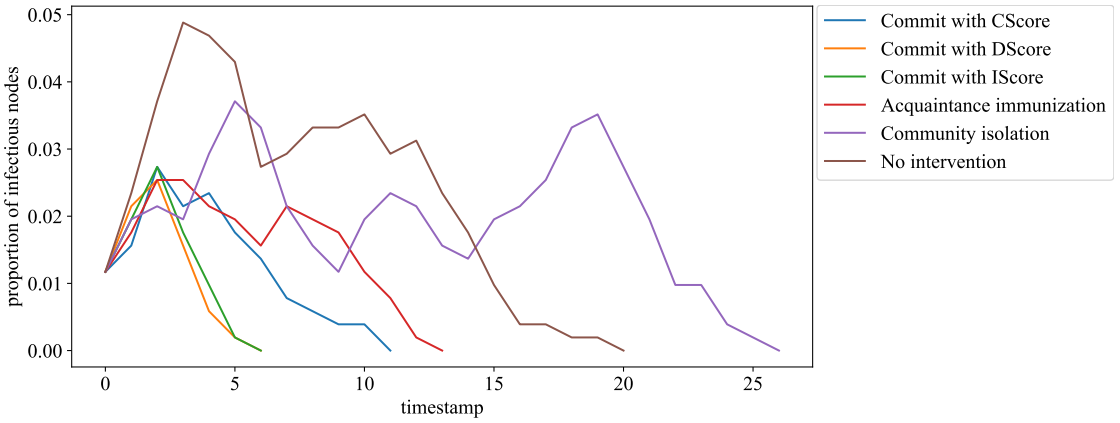


FIGURE 6.4: *The change in the dynamic of the spread due to mitigation strategies for Copenhagen dataset for SIR contagion model. Community isolation worsens the situation compared to no intervention scenario. All variants of ComMit outperform the benchmarks. Unlike the scenario with SIS model, community-based ComMit loses its advantage in terms of lowering the duration of infection compared to ComMit wiht DScore and IScore.*

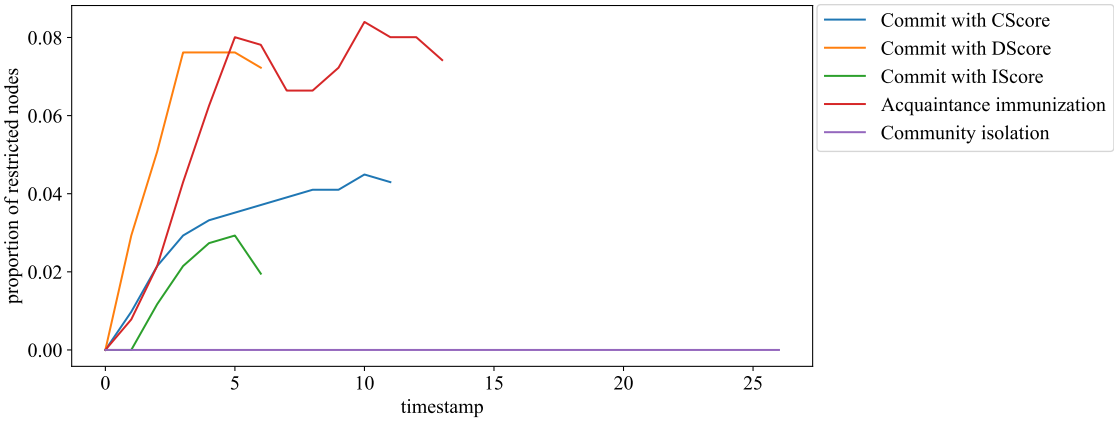


FIGURE 6.5: *The number of restricted nodes at each timestamp for Copenhagen dataset under SIR contagion model. The zero restriction of community isolation shows that the strategy has not been activated due to the small magnitude of contagion under SIR regime. ComMit with CScore and IScore require a smaller restriction magnitude, but ComMit with DScore and IScore yield the shortest absorption time. Under short-lived contagions such as those modeled by SIR, ComMit with CScore has a less pronounced advantage compared to the other two variants.*

		commit-cscore	commit-dscore	commit-iscore	acq-imm	com-iso	no-int
Albany	**budget**	0.056 (0.006)	0.064 (0.003)	0.043 (0.007)	0.086 (0.0)	0.83 (0.162)	NA
	duration	43 (9.8)	36.2 (8.4)	46.2 (7.3)	200.0 (0.0)	200.0 (0.0)	200.0 (0.0)
	inf_peak	0.051 (0.007)	0.045 (0.007)	0.047 (0.009)	0.136 (0.028)	0.18 (0.028)	0.186 (0.032)
Syracuse	**budget**	0.029 (0.003)	0.053 (0.004)	0.018 (0.005)	0.066 (0.001)	0.018 (0.053)	NA
	duration	19 (6.4)	12.9 (4.2)	24.3 (6.3)	194.0 (18.0)	200.0 (0.0)	200.0 (0.0)
	inf_peak	0.022 (0.003)	0.02 (0.002)	0.024 (0.004)	0.025 (0.004)	0.025 (0.005)	0.025 (0.004)
Rochester	**budget**	0.12 (0.024)	0.112 (0.016)	0.085 (0.004)	0.087 (0.001)	0.785 (0.087)	NA
	duration	61.7 (12.0)	69.0 (14.7)	88.4 (40.1)	200.0 (0.0)	200.0 (0.0)	200.0 (0.0)
	inf_peak	0.167 (0.062)	0.152 (0.042)	0.274 (0.117)	0.347 (0.079)	0.374 (0.092)	0.373 (0.091)
Copenhagen	**budget**	0.209 (0.011)	0.204 (0.013)	0.088 (0.0)	0.088 (0.001)	0.952 (0.024)	NA
	duration	49.2 (7.0)	58.4 (9.6)	200.0 (0.0)	200.0 (0.0)	200.0 (0.0)	200.0 (0.0)
	inf_peak	0.51 (0.032)	0.487 (0.042)	0.753 (0.017)	0.786 (0.012)	0.727 (0.044)	0.856 (0.004)
Ithaca	**budget**	0.117 (0.023)	0.098 (0.018)	0.057 (0.02)	0.068 (0.009)	0.476 (0.303)	NA
	duration	16.9 (9.1)	21.6 (7.1)	30.5 (23.1)	146.6 (82.2)	146.4 (82.2)	146.0 (82.7)
	inf_peak	0.121 (0.064)	0.113 (0.065)	0.138 (0.095)	0.202 (0.122)	0.211 (0.119)	0.209 (0.122)

TABLE 6.3: *Performance of various mitigation strategies for SIS model. Community-based and degree-based ComMit consistently reduce the peak of infection and the absorption time with limited budget, whereas the other methods do not give consistent performance gain across all datasets. The results are averaged among* 10 *runs of the simulation and the value in parenthesis shows the standard deviation.*

		commit_cscore	commit_dscore	commit_iscore	acq_imm	com_iso	no_mit
Albany	**budget**	0.03 (0.003)	0.057 (0.003)	0.012 (0.002)	0.076 (0.004)	0.0 (0.0)	NA
	duration	8.3 (1.1)	7.5 (1.7)	8.4 (1.9)	9.1 (1.3)	8.4 (1.20	8.8 (1.4)
	inf_peak	0.016 (0.001)	0.015 (0.002)	0.017 (0.001)	0.017 (0.002)	0.016 (0.002)	0.017 (0.002)
Syracuse	**budget**	0.017 (0.003)	0.05 (0.005)	0.007 (0.002)	0.045 (0.006)	0.0 (0.0)	NA
	duration	6.8 (1.7)	5.7 (1.2)	5.9 (1.5)	6.5 (1.8)	6.7 (1.4)	7.3 (1.005)
	inf_peak	0.013 (0.001)	0.013 (0.001)	0.012 (0.001)	0.014 (0.002)	0.013 (0.001)	0.013 (0.001)
Rochester	**budget**	0.054 (0.005)	0.063 (0.004)	0.027 (0.01)	0.084 (0.003)	0.0 (0.0)	NA
	duration	18.8 (5.9)	18.0 (5.6)	21.1 (7.4)	19.4 (7.4)	28.7 (8.5)	34.0 (7.0)
	inf_peak	0.029 (0.009)	0.026 (0.007)	0.031 (0.012)	0.034 (0.014)	0.039 (0.016)	0.045 (0.016)
Copenhagen	**budget**	0.062 (0.012)	0.074 (0.005)	0.034 (0.008)	0.081 (0.003)	0.088 (0.074)	NA
	duration	9.6 (4.2)	9.0 (2.4)	10.9 (3.8)	14.0 (5.4)	16.4 (6.6)	21.7 (7.0)
	inf_peak	0.028 (0.014)	0.027 (0.011)	0.039 (0.014)	0.037 (0.018)	0.042 (0.02)	0.052 (0.016)
Ithaca	**budget**	0.061 (0.014)	0.065 (0.008)	0.028 (0.027)	0.051 (0.023)	0.087 (0.173)	NA
	duration	5.8 (3.4)	5.5 (3.8)	5.0 (2.1)	6.9 (2.9)	6.3 (3.0)	6.2 (2.6)
	inf_peak	0.035 (0.022)	0.027 (0.014)	0.044 (0.038)	0.051 (0.037)	0.058 (0.046)	0.06 (0.048)

TABLE 6.4: *Performance of various mitigation strategies for SIR model. All variants of ComMit consistently reduce the peak of infection and the absorption time with limited budget, whereas the other methods do not give consistent performance gain across all datasets. The small magnitude of contagion under SIR regime makes the advantage of ComMit with CScore less pronounced compared to the other two variants. The results are averaged among* 10 *runs of the simulation and the value in parenthesis shows the standard deviation.*

The results show that all variants of *ComMit* outperform *Acquaintance Immunization*, *Community Isolation*, and *No Mitigation*. However, due to the short duration of infection with SIR model, we see that the advantages of *ComMit with CScore* are less pronounced compared to *ComMit with DScore* and *ComMit with IScore*. *ComMit with DScore* consistently yields the shortest absorption time out of the three, while overall *ComMit with CScore* does a better job at limiting the budget spent on the divider block. Although *ComMit with IScore* results in lower number of restricted nodes for some

datasets (see Figure 6.4), the result is not consistent across all datasets, as in the case with SIS model. Due to the small magnitude of the contagion under SIR model, the *Community Isolation* strategy is not activated for majority of datasets (see its zero budget for Albany, Syracuse, and Rochester in Table 6.4) as it does not meet the required threshold of infection.

6.5.2 Ablation Study on Test-Trace Block

Effectiveness of ϵ-greedy. *ComMit* uses ϵ-greedy algorithm for its test-trace block. Here, I justify this selection by comparing against alternative testing methods. My testing benchmarks include,

- **Random:** At each timestamp, select the pivot nodes randomly. Despite its simplicity, this method has been used extensively in practice (e.g., in the face of the COVID-19 pandemic).
- **Random with memory:** The same as Random testing, but the nodes tested in the previous timestamp are excluded. This strategy avoids the redundant testing of the recently visited nodes.
- **Degree with memory:** Select pivot nodes according to their degree centrality in the known graph, G_t^*. Exclude the nodes that have been tested in the previous timestamp.
- **ϵ-greedy:** As described in Section 6.3.1.
- **ϵ-memory:** Same as ϵ-greedy but excluding the nodes tested in the previous timestamp.
- **ϵ-degree:** Similar to ϵ-greedy but the policy for selection with $p(\epsilon_t)$ in Equation 6.5 changes to selection based on the degree centrality in the known graph, G_t^*.

Note that the tracing mechanism is the same across all testing strategies for fair comparison (see Section 6.3.1). I report the simulation result for the Copenhagen dataset under SIS contagion model below.

Figures 6.6 and 6.7 show how these testing methods compare in terms of efficiency and efficacy. The efficiency achieved by ϵ-greedy and ϵ-memory is comparable to that in

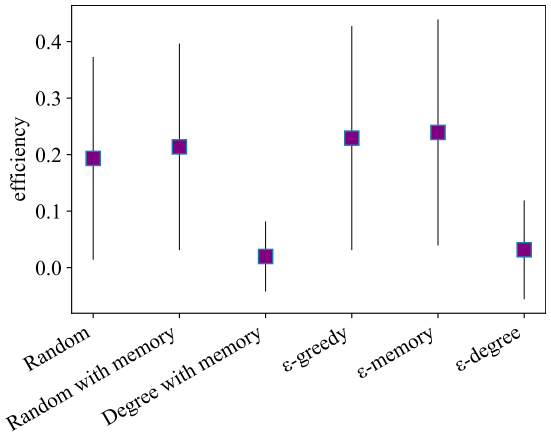


FIGURE 6.6: *The testing efficiency average and standard deviation for* 10 *simulation runs.* ϵ*-greedy and* ϵ*-memory yield the best efficiency, whereas degree-based methods give the worst.*

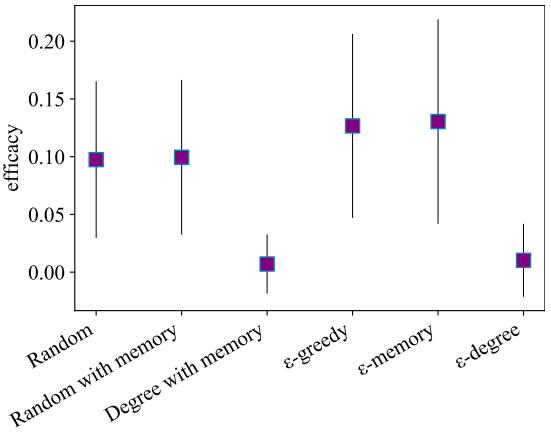


FIGURE 6.7: *The testing efficacy average and standard deviation for* 10 *simulation runs.* ϵ*-greedy and* ϵ*-memory yield the best efficacy, whereas degree-based methods give the worst.*

[166] whose testing data-driven testing method based on logistic regression and priority ranking on COVID-19 Lahore dataset produced 28.18 percent efficiency. The main difference between this work and theirs is that my testing strategy, in addition to finding infectious nodes, also reconstruct the contact network. The efficacy of the testing methods are lower than their efficiency, as expected. However, ϵ-greedy and ϵ-memory give a considerably higher performance and degree-based methods consistently perform poorly.

The graph-reconstruction power of each testing method, measured by the KL divergence between the degree distributions as explained in Section 6.4.3, is shown in Figure 6.8. The degree-based methods struggle to reconstruct the entirety of the graph within the duration of the spread. Other methods, reconstruct the full network within one forth of the total duration.

Combining the results of the efficiency, efficacy, and KL divergence measures, it is evident that the ϵ-greedy and ϵ-memory methods give the best performance for the exploration-exploitation task. The reason I chose the former for *ComMit* pipeline is the difference in their computation time. Since ϵ-memory has to remember the result from the previous timestamp, it is slower and does not yield a considerable boost in performance compare to the faster alternative, ϵ-greedy.

I also compare the final result of *ComMit* pipeline by using different testing methods

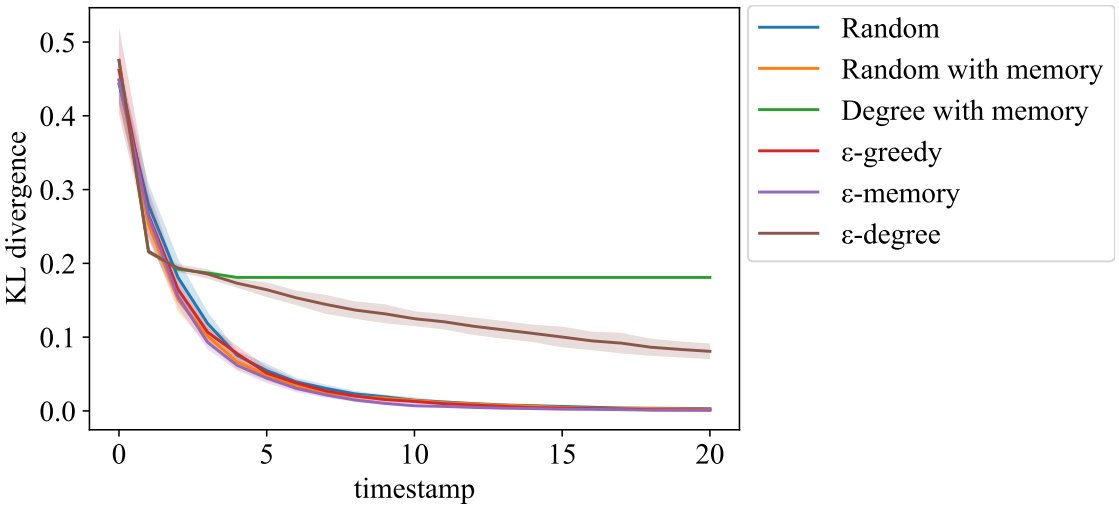


FIGURE 6.8: *Graph construction comparison. The KL divergence between degree distribution of the known (G_t^*) and unknown graphs (G_s). Degree-based methods fail to capture the entirely of the graph, whereas the other methods reconstruct the full graph within a short amount of time.*

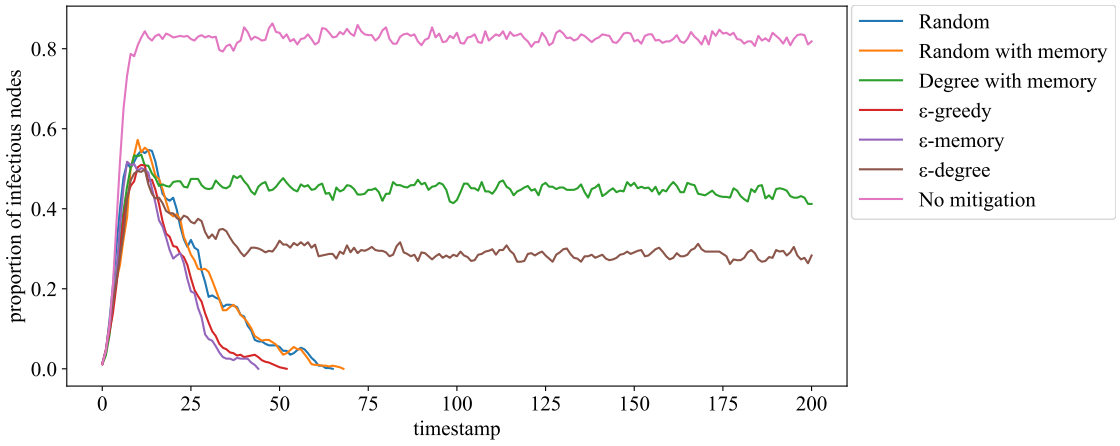


FIGURE 6.9: *The change in the dynamic of the spread due to testing strategies for Copenhagen dataset under SIS contagion model. ϵ-greedy and ϵ-memory give the best performance.*

and report them in Figures 6.9 and 6.10. Excluding the degree-based testing, all methods have comparable divider budget (6.10), whereas the lowest peak and duration of infection is obtained by ϵ-greedy and ϵ-memory (6.9).

Hyperparameters. The impact of self-reports accuracy, a_t, is tested in Figure 6.11 on the left. Higher accuracy results in discovering more edges quickly, but does not change the performance of *ComMit* drastically. This result suggests that *ComMit* does not rely on full knowledge of the graph to reach its best performance. In Figure 6.11 on the

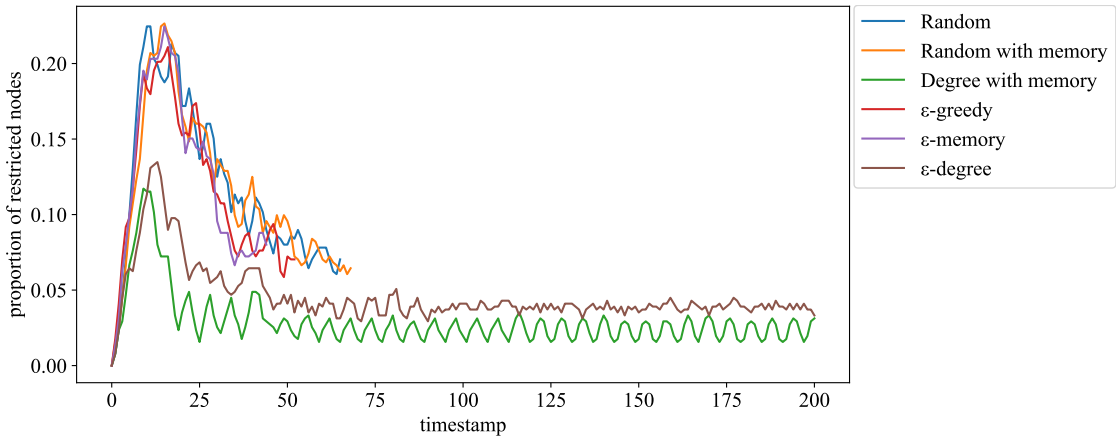


FIGURE 6.10: *The number of restricted nodes at each timestamp for Copenhagen dataset under SIS contagion mode. Excluding the degree-based testing, all methods have comparable divider budget.*

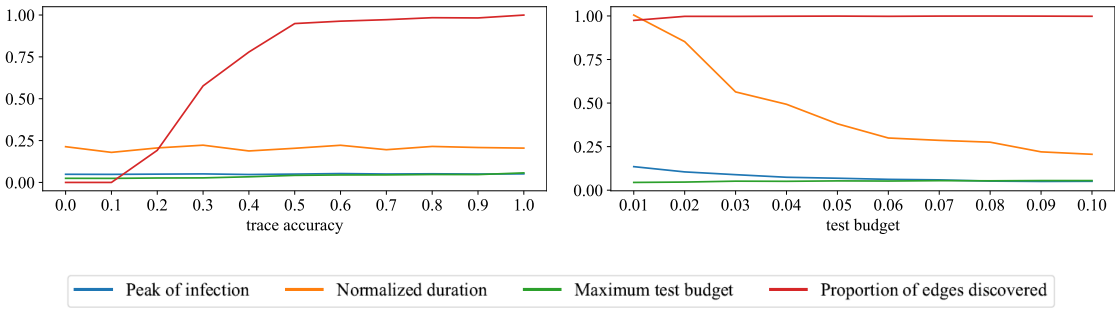


FIGURE 6.11: *Ablation study on trace accuracy (a_t) and test budget (b_t). The* `inf_peak`*,* `norm_duration`*,* `max_bud`*, and* `num_edges` *signify the peak of infection, the duration of infection normalized by the duration of simulation, the maximum divider budget in terms of number of restricted nodes normalized by $|V|$, and the number of edges discovered by the test strategy normalized by the number of edges in G_s, respectively.*

right, we see that increasing the test budget b_t, for $a_t = 1$, can drastically shorten the absorption time. However, small values of b_t still do a well at probing the full graph.

6.5.3 Ablation Study on Divider Block

The two top figures in 6.12 show that by increasing the divider's budget, b_{fs} and b_{fn}, no significant performance boost is observed. In the bottom figure of 6.12, I keep the duration of infection, t_d as 3 and change the divider's restriction time, t_r. The result shows that choosing a value closer to the actual infection time yields a shorter absorption time.

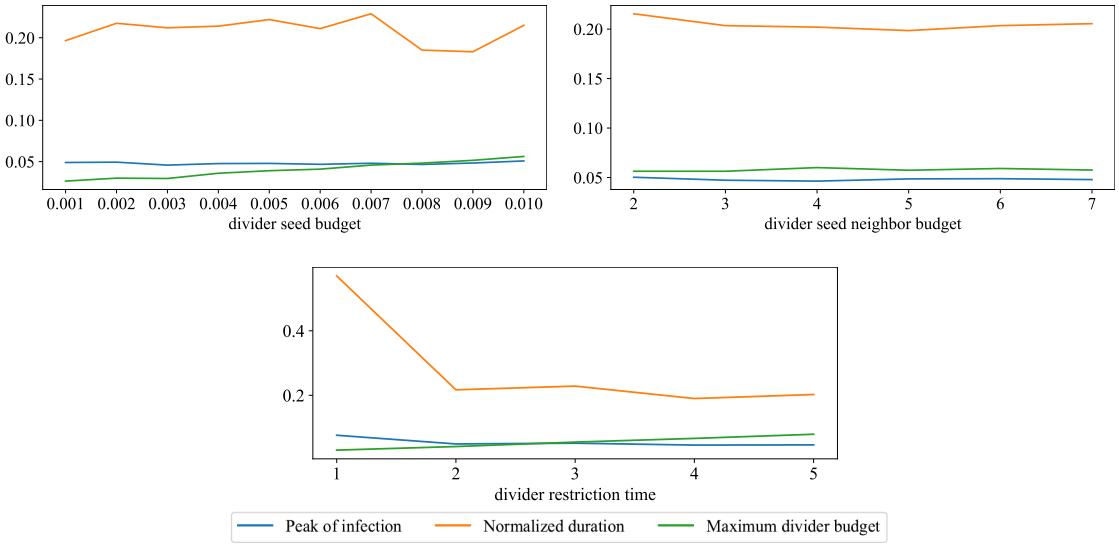


FIGURE 6.12: *Ablation study on divider seed budget (b_{fs}), divider seed neighbor budget (b_{fn}), and divider restriction time (t_r) in comparison with the disease duration (3). The* `inf_peak`*,* `norm_duration`*, and* `max_bud` *signify the peak of infection, the duration of infection normalized by the duration of simulation, and the maximum divider budget in terms of number of restricted nodes normalized by |V|, respectively.*

6.5.4 Blind vs. Non-Blind Performance

What distinguishes *ComMit* from similar mitigation strategies is its blindness assumption on the contact network structure. I empirically demonstrated the effectiveness of *ComMit* in Section 6.5.1. Here, my goal is to obtain the same results but under the non-blind assumption; i.e., the algorithm knows the graph structure G_s a priori. This removes the necessity for test-trace block in Figure 6.1 and reduces the algorithm to an iterative divider block.

Figures 6.13 and 6.14 show the performance and budget of different mitigation strategies under non-blind assumption for SIS contagion model and Copenhagen dataset. Comparing these two figures with those in 6.2 and 6.3, we see that the results are exactly the same in terms of how different strategies compare. If we consider *ComMit* with CScore and DScore, however, we notice that the DScore has lost its advantage over CScore by having and increased peak of infection. CScore, on the other hand, has a lower peak compared to blind scenario. As expected, both reach absorption time faster under non-blind assumption. However, their performance is still comparable under blind and non-blind assumptions, which shows the success of *ComMit*'s testing strategy in alleviating the blindness limitation.

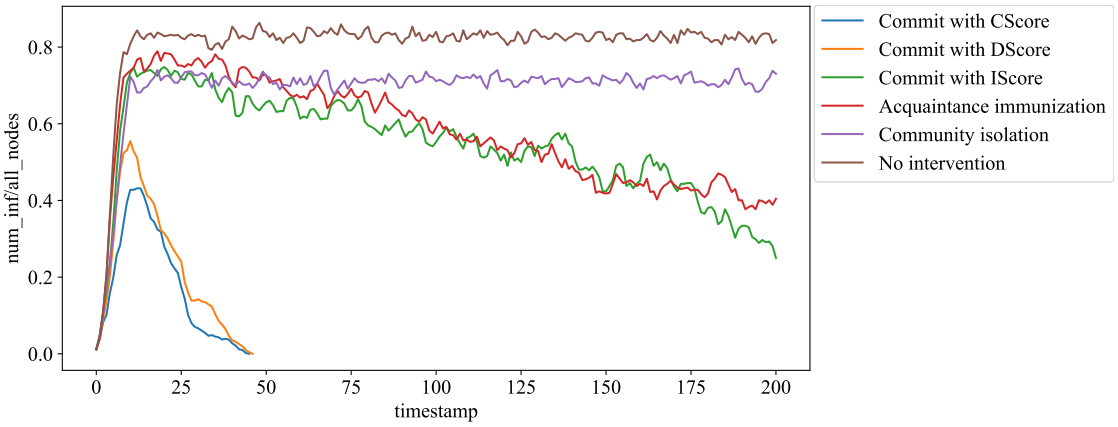


FIGURE 6.13: *The change in the dynamic of the spread for Copenhagen dataset under non-blind assumption and SIS contagion model. Both ComMit with CScore and Dscore reach the absorption time faster than that under the blindness assumption (Figure 6.2). CScore gives a lower peak of infection than DScore in this scenario.*

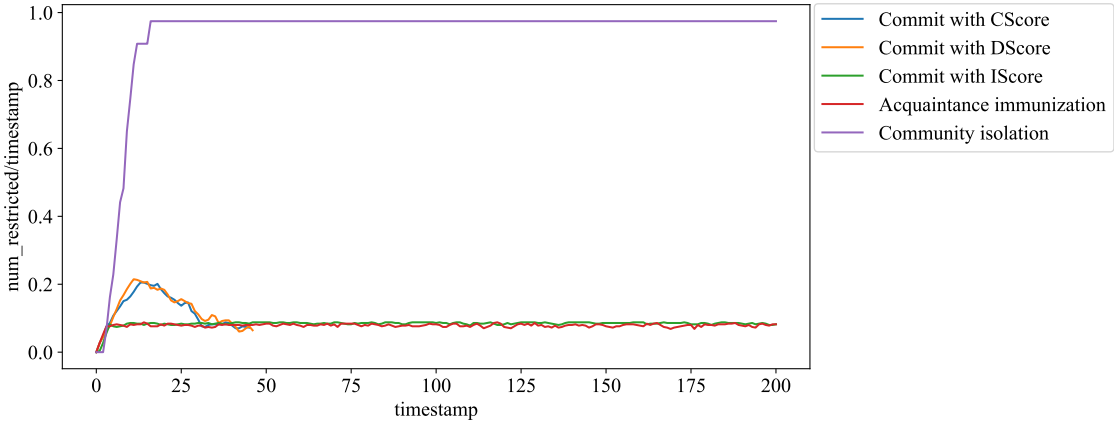


FIGURE 6.14: *The number of restricted nodes at each timestamp for Copenhagen dataset under non-blind assumption and SIS contagion model. The results are similar to those in Figure 6.3, which shows the blindness limitation does not increase the required budget for ComMit.*

6.5.5 Performance under High Infection Rate

Blindness to the dynamics of the contagion is one of the requirements for a practical early mitigation strategy. An ideal strategy should not lose its capability in inhibiting the contagion as the contagion rate (i.e., infection rate) goes up. For such a model, we expect the model's performance to reach its stable performance in a relatively small infection rate and becomes agnostic to higher rates. Using the contagion metrics – absorption time and peak of infection – I compare the performance of *ComMit* with the benchmarks under increasing infection rates for SIS contagion model. The results for Albany dataset is shown in Figure 6.15.

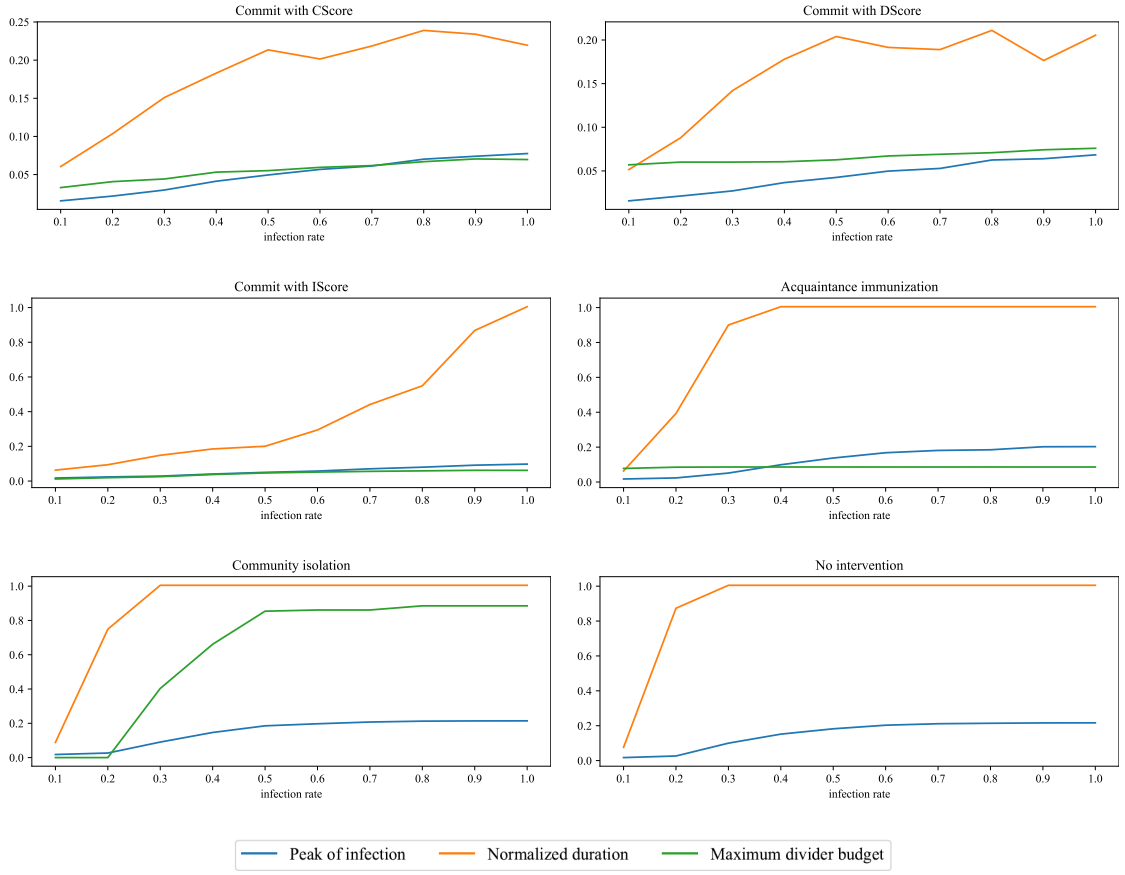


FIGURE 6.15: *Performance comparison for different infection rates in SIS contagion model. The* `inf_peak`*,* `norm_duration`*, and* `max_bud` *signify the peak of infection, the duration of infection normalized by the duration of simulation, and the maximum divider budget in terms of number of restricted nodes normalized by* $|V|$*, respectively. ComMit with CScore and DScore are the only strategies whose performance is not disturbed by the higher values of the infection rate. These two models give the best trade-off between budget and performance as well.*

ComMit with CScore and Dscore are the only strategies whose performance is not disturbed by higher infection rates. *ComMit* with IScore, despite maintaining a low peak of infection, steadily loses its ability in controlling the duration of the spread for higher rates. Both acquaintance immunization and community isolation lose their performance in smaller values of infection rate and reach the state of the spread with no intervention for higher values. I also have shown the divider budget for each strategy (as the testing budget is the same across all) in the same figure. It shows that lower budgets are not always the indication of a better mitigation strategy (e.g., *ComMit* with IScore) and the best trade-off between the budget and performance is achieved by *ComMit* with CScore and DScore.

6.6 Summary

In this chapter, I formally defined the problem of early mitigation strategy and offered a dynamic algorithm, *ComMit*, that incorporates the realistic assumptions of blindness towards network and spread dynamics. More specifically, *ComMit* addresses the problem of early mitigation strategy from a *community perspective*, and has two main features distinguishing it from prior work: (1) It is agnostic to the dynamics of the spread; (2) does not require prior knowledge on contact network; (3) it works within a limited budget; and (4) it enforces bursts of short-term restriction on small communities instead of long-term isolation of healthy individuals.

ComMit only utilizes geographical information to infer community membership and data from testing and contact tracing to update its knowledge on the spread without enforcing any assumptions about the nature of the disease. Because *ComMit* relies on updated data from test-trace reports, it is dynamic and its proposed mitigation strategy can evolve over time. Using the updated information, it introduces network perturbations that control the magnitude of the spread by following a community fragmentation strategy.

I tested *ComMit* on several real-world social networks. The results of the experiments in this chapter show that, within a small budget, *ComMit* can reduce the peak of infection by 73% and shorten the duration of infection by 90%, even for spreads that would reach a steady state of non-zero infections otherwise (e.g., SIS contagion model).

It is worth noting that *ComMit* relies on geo-network only for estimating the community structure in the contact network. If these communities are known through other means (e.g., government survey data), no geo information is required.

Limitations & Future Direction. In this chapter, I have not considered the scenario of multiple pandemics co-occurring in the network (e.g., different mutations of COVID-19). I hypothesize that under such circumstances, *ComMit* still approaches to the absorption state but in a longer time frame. Another assumption that is not covered here is the non-ideal testing kit with less than 100% accuracy in determining if the tested candidate is infectious or not. In the case of more sophisticated contagion models, such as those with varying infection parameters [167], it is expected that *ComMit* maintains its performance as it does not rely on a priori information on contagion dynamics. However, this hypothesis has not been tested.

Throughout this chapter, I assumed the speed of time varying graph compared to the dynamics of a disease is much slower. In majority of viral infection, this is indeed the case. However, there is an alternative scenario in which the two speeds are comparable (e.g., as studied by Nadini et al. [168]). In this case, the updated observation of contact layer in *ComMit* needs to be readjusted with respect to the varying dynamics of the network. In general, the absence of contact data for the dynamic contact networks is a more complex problem that deserves its own separate study.

Ethical Issues. Owing to the exploration component, it is possible to test a candidate with low infection probability. There are ethical issues involved with violating one's privacy by requiring their social information when they are not likely to put others in danger.

Chapter 7

Conclusion

In real-world complex networks, destructive spreads, commonly known as contagions, are common and can potentially lead to catastrophic events if uncontrolled. Some examples are biological contagions such as pandemics that disturb societies and potentially lead to death of many; or network attack contagions on crucial infrastructure systems, such as power grids and water treatments, that disrupt societies as a whole; or social contagions such as the propagation of misinformation or radical ideas, ensuing chaos and polarization within and across societies. For these reasons, it is critical to study the protective measures against contagions in complex networks.

In this dissertation, I studied the network protection problem in the context of network attacks and biological contagions. The outcome of this effort makes several important contributions to the network protection field, as summarized below.

I first started by reviewing fundamental graph theoretic concepts and related work in network spreading processes, network protection research, and contagions (Chapter 2). By organizing the bulk of research in contagions into three categories – network attacks, biological, and social contagions – I highlighted the common major shortcomings in the current network protection trends; namely, using global graph knowledge, ignoring dynamic nature of spread and network, imprecise evaluation metrics, and scalability.

In Chapter 3 and to address the problem of scalability, I proved, theoretically and empirically, the existence of a relationship between characteristic path length and local clustering coefficient. The characteristic path length (i.e., the average of all pairwise shortest paths) is a global network measure that directly impacts the contagion paths (i.e, the paths through which the spread propagates). The expensive computation cost

of this measure makes it an impractical tool in designing protection algorithms for real-world settings. On the other hand, the local clustering coefficient is relatively fast to compute. This local measure indicates the strength of a cluster (community) formed by the neighbors of a node. The relationship between the two suggests that, instead of global network manipulations, we can disrupt the contagion pathways by manipulating the local community of "certain nodes".

Depending on the problem setting, these "certain nodes" are identified in two different ways: (1) predefined critical nodes based on the sensitive information they carry, or their overall importance to the functionality of the network (Chapter 4); (2) unknown critical nodes whose importance depends on their location in the network structure and interaction with other nodes (e.g., in a community), and have to be detected based on the current network structure at each time (Chapter 6).

In Chapter 4, I focused on network attack contagions and defined the problem of protecting a set of (predefined) target nodes against an unknown intruding contagion. I proposed the *CoVerD* algorithm in this chapter which only uses the local community information of the target nodes. Tested on real networks and compared with existing methods, *CoVerD* achieved the lowest closeness centrality for target nodes without using any global measures. While maintaining the fast computation of local network perturbations, it improved the best performing benchmarks (some of which were based on global measures, such as betweenness centrality) by increasing the attacker's required budget (i.e., the effort required for contagion to reach the target nodes) by 3 to 10 times. This chapter emphasized the importance of choosing a proper evaluation metric (the attack budget vs. global centrality measures) and using local community information to enhance scalability.

In Chapter 5, I turned to biological contagions and studied the network protection problem in early stages of an unknown viral spread. In this problem setting, the set of nodes that need to be protected are not predefined and the protection algorithm needs a ranking method for identifying the most critical ones. It was revealed that, when considering both the magnitude of the spread and cost of the protection strategy, the node ranking based on the $1 - hop$ information is enough to obtain the best trade-off. The results are compatible with those obtained in the prior chapter on importance of the local information and proper evaluation metric. The analysis in this chapter, however, was based on the ideal assumption that we have access to the contact information of all nodes in the graph.

In Chapter 6, I removed the ideal assumption of full knowledge on the contact information and introduced the problem of *blind* network protection. This problem is specifically of importance for real-world scenarios in which policy makers have to make urgent decisions in the early stages of a dangerous spread. I formalized this problem and designed a dynamic mitigation algorithm, *ComMit*, that successfully terminates the persistent contagions fast and within a limited budget. The dynamic nature of *ComMit* allows for simultaneous information gathering through test-trace procedure and introducing the network modifications that, to the best of the algorithm's knowledge at each time, efficiently inhibits the unknown contagion. The effectiveness of *ComMit* was shown on real-world data where it reduced the peak of infection by 73% and shortened the duration of spread by 90%.

ComMit is the first attempt in addressing the blindness to both the network structure and contagion dynamics, and the outstanding performance of *ComMit* shows a promising path forward. The major implication of these results is the possibility of devising practical mitigation policies in the face of viral emergencies that do not drastically disrupt the societies (in contrast with, for example, lockdown and herd-immunity-based policies).

The attempt in this dissertation has been to encourage further research in the area of network protection against contagions that consider real-world limitations. To this end, I argue that there needs to be a shift from using global network measures to local ones (such as local community information) to improve the scalability of the protection methods. I showed that the accessibility of a node or group of nodes is best controlled by considering their local neighborhood information rather than full graph structure, which is often not available. Even under the strict limitations of real-world scenarios, such as dealing with unknown graph structure and unknown contagion dynamics, it is possible to design mitigation strategies that are fast, effective, and relatively easy to implement by resourceful policy makers.

There are several interesting directions to continue this work. To name a few, one is to consider the multi-contagion scenario in which several spreading processes navigate the network simultaneously. This is a common scenario in real networks; e.g., the existence of different variants of a disease in a population, or parallel attacks on an infrastructure to increase the potential of damage. It might also be of interest, at least theoretically, to consider contagions with changing dynamics. For instance, diseases with varying infection rate, or smart crawling attackers that update the crawling policy based on the information they gather. Possibly the most compelling direction is to analyze the

results of this research from a public policy perspective to find possible bottlenecks and improvements in the proposed protection algorithms, which can save the lives of many.

Bibliography

[1] Duncan J Watts and Steven H Strogatz. Collective dynamics of small-world networks. *nature*, 393(6684):440, 1998.

[2] Filippo Aureli and Gabriele Schino. Social complexity from within: how individuals experience the structure and organization of their groups. *Behavioral Ecology and sociobiology*, 73(1):6, 2019.

[3] John Adrian Bondy, Uppaluri Siva Ramachandra Murty, et al. *Graph theory with applications*, volume 290. Macmillan London, 1976.

[4] Reinhard Diestel. Graph theory 3rd ed. *Graduate texts in mathematics*, 173:33, 2005.

[5] Reza Zafarani, Mohammad Ali Abbasi, and Huan Liu. *Social media mining: an introduction*. Cambridge University Press, 2014.

[6] Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3-5), 2010.

[7] Santo Fortunato and Mark E.J. Newman. 20 years of network community detection. *Nature Physics*, 2022.

[8] Ana Maria Hernandez-Hernandez, Dolores Viga-de Alva, Rodrigo Huerta-Quintanilla, Efrain Canto-Lugo, Hugo Laviada-Molina, and Fernanda Molina-Segui. Friendship concept and community network structure among elementary school and university students. *PloS one*, 11(10):e0164886, 2016.

[9] Ayalvadi Ganesh, Laurent Massoulié, and Don Towsley. The effect of network topology on the spread of epidemics. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, volume 2, pages 1455–1466. IEEE, 2005.

[10] Dong Wang, Michael Small, and Yi Zhao. Exploring the optimal network topology for spreading dynamics. *Physica A: Statistical Mechanics and its Applications*, 564:125535, 2021.

[11] Augusto Santos and José MF Moura. Diffusion and topology: Large densely connected bipartite networks. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 2738–2743. IEEE, 2012.

[12] Jie Zhou, Zonghua Liu, and Baowen Li. Influence of network structure on rumor propagation. *Physics Letters A*, 368(6):458–463, 2007.

[13] Heejo Lee and Jong Kim. Attack resiliency of network topologies. In *International Conference on Parallel and Distributed Computing: Applications and Technologies*, pages 638–641. Springer, 2004.

[14] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[15] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.

[16] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006.

[17] Roger Guimera, Marta Sales-Pardo, and Luís A Nunes Amaral. Modularity from fluctuations in random graphs and complex networks. *Physical Review E*, 70(2): 025101, 2004.

[18] Liangxun Shuo and Bianfang Chai. Discussion of the community detection algorithm based on statistical inference. *Perspectives in Science*, 7:122–125, 2016.

[19] Liudmila Prokhorenkova and Alexey Tikhonov. Community detection through likelihood optimization: in search of a sound model. In *The World Wide Web Conference*, pages 1498–1508, 2019.

[20] Ludvig Bohlin, Daniel Edler, Andrea Lancichinetti, and Martin Rosvall. Community detection and visualization of networks with the map equation framework. In *Measuring scholarly impact*, pages 3–34. Springer, 2014.

[21] Martin Rosvall and Carl T Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the national academy of sciences*, 105(4):1118–1123, 2008.

[22] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12): 7821–7826, 2002.

[23] Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):1–12, 2019.

[24] Marcin Waniek, Tomasz P Michalak, Michael J Wooldridge, and Talal Rahwan. Hiding individuals and communities in a social network. *Nature Human Behaviour*, 2(2):139–147, 2018.

[25] Marcel Salathé and James H Jones. Dynamics and control of diseases in networks with community structure. *PLoS Comput Biol*, 6(4):e1000736, 2010.

[26] Olle Abrahamsson. Hide and seek in a social network. Master's thesis, Linköping University, Sweden, 2017.

[27] Pierluigi Crescenzi, Gianlorenzo d'Angelo, Lorenzo Severini, and Yllka Velaj. Greedily improving our own centrality in a network. In *International Symposium on Experimental Algorithms*, pages 43–55. Springer, 2015.

[28] Christian M Schneider, Tamara Mihaljev, Shlomo Havlin, and Hans J Herrmann. Suppressing epidemics with a limited amount of immunization units. *Physical Review E*, 84(6):061911, 2011.

[29] Jie Ji, Guohua Wu, Chenjian Duan, Yizhi Ren, and Zhen Wang. Greedily remove k links to hide important individuals in social network. In *International Symposium on Security and Privacy in Social Networks and Big Data*, pages 223–237. Springer, 2019.

[30] Neda Jahanshad, Gautam Prasad, Arthur W Toga, Katie L McMahon, Greig I de Zubicaray, Nicholas G Martin, Margaret J Wright, and Paul M Thompson. Genetics of path lengths in brain connectivity networks: Hardi-based maps in 457 adults. In *International Workshop on Multimodal Brain Image Analysis*, pages 29–40. Springer, 2012.

[31] Robin Cowan and Nicolas Jonard. Network structure and the diffusion of knowledge. *Journal of economic Dynamics and Control*, 28(8):1557–1575, 2004.

[32] Thibaud Trolliet, Nathann Cohen, Frédéric Giroire, Luc Hogie, and Stéphane Pérennes. Interest clustering coefficient: a new metric for directed networks like twitter. *Journal of Complex Networks*, 10(1):cnab030, 2022.

[33] Roy M Anderson. Discussion: the kermack-mckendrick epidemic threshold theorem. *Bulletin of mathematical biology*, 53(1):1–32, 1991.

[34] Mark Granovetter. Threshold models of collective behavior. *American journal of sociology*, 83(6):1420–1443, 1978.

[35] Mark Granovetter and Roland Soong. Threshold models of diffusion and collective behavior. *Journal of Mathematical sociology*, 9(3):165–179, 1983.

[36] Valeriano Iranzo and Saúl Pérez-González. Epidemiological models and covid-19: a comparative view. *History and Philosophy of the Life Sciences*, 43(3):1–24, 2021.

[37] Duncan J Watts. A simple model of global cascades on random networks. *Proceedings of the National Academy of Sciences*, 99(9):5766–5771, 2002.

[38] Jon Kleinberg. Cascading behavior in networks: Algorithmic and economic issues. *Algorithmic game theory*, 24:613–632, 2007.

[39] Paul L Krapivsky and Sidney Redner. Dynamics of majority rule in two-state interacting spin systems. *Physical Review Letters*, 90(23):238701, 2003.

[40] B Pittel. On a daley-kendall model of random rumours. *Journal of Applied Probability*, 27(1):14–27, 1990.

[41] Stephen Morris. Contagion. *The Review of Economic Studies*, 67(1):57–78, 2000.

[42] Romualdo Pastor-Satorras and Alessandro Vespignani. Epidemic spreading in scale-free networks. *Physical review letters*, 86(14):3200, 2001.

[43] Maksim Kitsak, Lazaros K Gallos, Shlomo Havlin, Fredrik Liljeros, Lev Muchnik, H Eugene Stanley, and Hernán A Makse. Identification of influential spreaders in complex networks. *Nature physics*, 6(11):888–893, 2010.

[44] Konstantin Klemm, M Serrano, Víctor M Eguíluz, and Maxi San Miguel. A measure of individual role in collective dynamics. *Scientific reports*, 2(1):1–8, 2012.

[45] Sen Pei, Lev Muchnik, José S Andrade Jr, Zhiming Zheng, and Hernán A Makse. Searching for superspreaders of information in real-world social media. *Scientific reports*, 4(1):1–12, 2014.

[46] Matteo Serafino, Higor S Monteiro, Shaojun Luo, Saulo DS Reis, Carles Igual, Antonio S Lima Neto, Matías Travizano, José S Andrade Jr, and Hernán A Makse. Superspreading k-cores at the center of covid-19 pandemic persistence. *arXiv preprint arXiv:2103.08685*, 2021.

[47] David Lando and Mads Stenbo Nielsen. Correlation in corporate defaults: Contagion or conditional independence? *Journal of Financial Intermediation*, 19(3): 355–372, 2010.

[48] Carlo A Favero and Francesco Giavazzi. Looking for contagion: Evidence from the erm, 2000.

[49] Christian Barrot and Sönke Albers. Did they tell their friends?-using social network analysis to detect contagion processes. *Using Social Network Analysis to Detect Contagion Processes (February 2008)*, 2008.

[50] Eli A Meirom, Chris Milling, Constantine Caramanis, Shie Mannor, Sanjay Shakkottai, and Ariel Orda. Localized epidemic detection in networks with overwhelming noise. *ACM SIGMETRICS Performance Evaluation Review*, 43(1): 441–442, 2015.

[51] Chris Milling, Constantine Caramanis, Shie Mannor, and Sanjay Shakkottai. Local detection of infections in heterogeneous networks. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 1517–1525. IEEE, 2015.

[52] Peter Sheridan Dodds. A simple person's approach to understanding the contagion condition for spreading processes on generalized random networks. In *Complex Spreading Phenomena in Social Systems*, pages 27–45. Springer, 2018.

[53] Duncan J Watts. A simple model of global cascades on random networks. In *The Structure and Dynamics of Networks*, pages 497–502. Princeton University Press, 2011.

[54] Ricky Laishram, Pegah Hozhabrierdi, Jeremy Wendt, and Sucheta Soundarajan. Netprotect: Network perturbations to protect nodes against entry-point attack. In *13th ACM Web Science Conference 2021*, pages 93–101, 2021.

[55] Ellsworth Campbell and Marcel Salathé. Complex social contagion makes networks more vulnerable to disease outbreaks. *Scientific reports*, 3(1):1–6, 2013.

[56] Dimitar Nikolov, Alessandro Flammini, and Filippo Menczer. Right and left, partisanship predicts (asymmetric) vulnerability to misinformation. *arXiv preprint arXiv:2010.01462*, 2020.

[57] Jure Leskovec, Lada A Adamic, and Bernardo A Huberman. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)*, 1(1):5–es, 2007.

[58] Brian Oldenburg and Karen Glanz. Diffusion of innovations. *Health behavior and health education: Theory, research, and practice*, 4:313–333, 2008.

[59] Antonio Caliò, Roberto Interdonato, Chiara Pulice, and Andrea Tagarelli. Topology-driven diversity for targeted influence maximization with application to user engagement in social networks. *IEEE Transactions on Knowledge and Data Engineering*, 30(12):2421–2434, 2018.

[60] John E Fontecha, Jose L Walteros, and Alexander Nikolaev. Reach maximization for social lotteries. *Omega*, page 102496, 2021.

[61] Pratha Sah, Stephan T Leu, Paul C Cross, Peter J Hudson, and Shweta Bansal. Unraveling the disease consequences and mechanisms of modular structure in animal social networks. *Proceedings of the National Academy of Sciences*, 114 (16):4165–4170, 2017.

[62] Bnaya Gross and Shlomo Havlin. Epidemic spreading and control strategies in spatial modular network. *Applied Network Science*, 5(1):1–14, 2020.

[63] Fabiana Zollo and Walter Quattrociocchi. Misinformation spreading on facebook. In *Complex spreading phenomena in social systems*, pages 177–196. Springer, 2018.

[64] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146, 2003.

[65] Douglas Guilbeault, Joshua Becker, and Damon Centola. Complex contagions: A decade in review. *Complex spreading phenomena in social systems*, pages 3–25, 2018.

[66] Saleh Soltan, Dorian Mazauric, and Gil Zussman. Cascading failures in power grids: analysis and algorithms. In *Proceedings of the 5th international conference on Future energy systems*, pages 195–206, 2014.

[67] Federica Agosta, Marina Weiler, and Massimo Filippi. Propagation of pathology through brain networks in neurodegenerative diseases: from molecules to clinical phenotypes. *CNS neuroscience & therapeutics*, 21(10):754–767, 2015.

[68] Peter Sheridan Dodds. Slightly generalized contagion: Unifying simple models of biological and social spreading. In *Complex Spreading Phenomena in Social Systems*, pages 67–80. Springer, 2018.

[69] Joseph Dureau, Konstantinos Kalogeropoulos, and Marc Baguelin. Capturing the time-varying drivers of an epidemic using stochastic dynamical systems. *Biostatistics*, 14(3):541–555, 2013.

[70] Jonathan S Yedidia, William T Freeman, Yair Weiss, et al. Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium*, 8:236–239, 2003.

[71] Jessie EC Adriaense, Jordan S Martin, Martina Schiestl, Claus Lamm, and Thomas Bugnyar. Negative emotional contagion and cognitive bias in common ravens (corvus corax). *Proceedings of the National Academy of Sciences*, 116 (23):11547–11552, 2019.

[72] Joyce E Bono and Remus Ilies. Charisma, positive emotions and mood contagion. *The Leadership Quarterly*, 17(4):317–334, 2006.

[73] Damián H Zanette. Dynamics of rumor propagation on small-world networks. *Physical review E*, 65(4):041908, 2002.

[74] Dumitru-Clementin Cercel and Stefan Trausan-Matu. Opinion propagation in online social networks: A survey. In *Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14)*, pages 1–10, 2014.

[75] Fang Jin, Wei Wang, Liang Zhao, Edward Dougherty, Yang Cao, Chang-Tien Lu, and Naren Ramakrishnan. Misinformation propagation in the age of twitter. *Computer*, 47(12):90–94, 2014.

[76] Zhiping Xiao, Weiping Song, Haoyan Xu, Zhicheng Ren, and Yizhou Sun. Timme: Twitter ideology-detection via multi-task multi-relational embedding. In

Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 2258–2268, 2020.

[77] Husheng Li, Ju Bin Song, Chien-fei Chen, Lifeng Lai, and Robert C Qiu. Behavior propagation in cognitive radio networks: A social network approach. *IEEE transactions on wireless communications*, 13(2):646–657, 2014.

[78] Mark S Granovetter. The strength of weak ties. *American journal of sociology*, 78(6):1360–1380, 1973.

[79] Stanley Milgram. The small world problem. *Psychology today*, 2(1):60–67, 1967.

[80] Rogier Noldus and Piet Van Mieghem. Assortativity in complex networks. *Journal of Complex Networks*, 3(4):507–542, 2015.

[81] Jessica Hoffmann and Constantine Caramanis. Learning graphs from noisy epidemic cascades. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 3(2):1–34, 2019.

[82] Samuel F Rosenblatt, Jeffrey A Smith, G Robin Gauthier, and Laurent Hébert-Dufresne. Immunization strategies in networks with missing data. *PLoS computational biology*, 16(7):e1007897, 2020.

[83] Michele Starnini, Andrea Baronchelli, and Romualdo Pastor-Satorras. Modeling human dynamics of face-to-face interaction networks. *Physical review letters*, 110(16):168701, 2013.

[84] Heather Mattie, Kenth Engø-Monsen, Rich Ling, and Jukka-Pekka Onnela. Understanding tie strength in social networks using a local "bow tie" framework. *Scientific reports*, 8(1):1–9, 2018.

[85] Hua Liang, Hongyu Miao, and Hulin Wu. Estimation of constant and time-varying dynamic parameters of hiv infection in a nonlinear differential equation model. *The annals of applied statistics*, 4(1):460, 2010.

[86] V Paul Poteat, Ethan H Mereish, Marcia L Liu, and J Sophia Nam. Can friendships be bipartisan? the effects of political ideology on peer relationships. *Group Processes & Intergroup Relations*, 14(6):819–834, 2011.

[87] János Kertész. Temporal networks: Characterization, motifs and spreading. *Computational Social Science and Complex Systems*, 203:105, 2019.

[88] Sen Pei and Hernán A Makse. Spreading dynamics in complex networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2013(12):P12002, 2013.

[89] Mark DF Shirley and Steve P Rushton. The impacts of network topology on disease spread. *Ecological Complexity*, 2(3):287–299, 2005.

[90] Eduardo R Pinto, Erivelton G Nepomuceno, and Andriana SLO Campanharo. Impact of network topology on the spread of infectious diseases. *TEMA (São Carlos)*, 21:95–115, 2020.

[91] Ruixia Zhang and Deyu Li. Rumor propagation on networks with community structure. *Physica A: Statistical Mechanics and its Applications*, 483:375–385, 2017.

[92] Eytan Katzav, Ofer Biham, and Alexander K Hartmann. Distribution of shortest path lengths in subcritical erdős-rényi networks. *Physical Review E*, 98(1): 012301, 2018.

[93] Yale Chai, Chunyao Song, Peng Nie, Xiaojie Yuan, and Yao Ge. Community structure based shortest path finding for social networks. In *International Conference on Database and Expert Systems Applications*, pages 303–319. Springer, 2018.

[94] Andrey Gubichev, Srikanta Bedathur, Stephan Seufert, and Gerhard Weikum. Fast and accurate estimation of shortest paths in large graphs. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 499–508. ACM, 2010.

[95] Michalis Potamias, Francesco Bonchi, Carlos Castillo, and Aristides Gionis. Fast shortest path distance estimation in large networks. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 867–876. ACM, 2009.

[96] Damon Centola. The spread of behavior in an online social network experiment. *science*, 329(5996):1194–1197, 2010.

[97] Jon Kleinberg. The small-world phenomenon: An algorithmic perspective. Technical report, Cornell University, 1999.

[98] Wayne Zachary. An information flow model for conflict and fission in small groups. *J. of Anthropological Research*, 33:452–473, 1977.

[99] Train bombing network dataset – KONECT, April 2017. URL http://konect.uni-koblenz.de/networks/moreno_train.

[100] Residence hall network dataset – KONECT, April 2017. URL http://konect.uni-koblenz.de/networks/moreno_oz.

[101] Haggle network dataset – KONECT, April 2017. URL http://konect.uni-koblenz.de/networks/contact.

[102] Infectious network dataset – KONECT, April 2017. URL http://konect.uni-koblenz.de/networks/sociopatterns-infectious.

[103] Hamsterster full network dataset – KONECT, April 2017. URL http://konect.uni-koblenz.de/networks/petster-hamster.

[104] Adolescent health network dataset – KONECT, April 2017. URL http://konect.uni-koblenz.de/networks/moreno_health.

[105] Facebook (nips) network dataset – KONECT, April 2017. URL http://konect.uni-koblenz.de/networks/ego-facebook.

[106] Advogato network dataset – KONECT, April 2017. URL http://konect.uni-koblenz.de/networks/advogato.

[107] Pretty good privacy network dataset – KONECT, April 2017. URL http://konect.uni-koblenz.de/networks/arenas-pgp.

[108] Qawi K Telesford, Karen E Joyce, Satoru Hayasaka, Jonathan H Burdette, and Paul J Laurienti. The ubiquity of small-world networks. *Brain connectivity*, 1 (5):367–375, 2011.

[109] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[110] Bob G Schlicher, Lawrence P MacIntyre, and Robert K Abercrombie. Towards reducing the data exfiltration surface for the insider threat. In *2016 49th Hawaii International Conference on System Sciences (HICSS)*. IEEE, 2016.

[111] Raïssa Yapan Dougnon, Philippe Fournier-Viger, and Roger Nkambou. Inferring user profiles in online social networks using a partial social graph. In *Canadian Conference on Artificial Intelligence*. Springer, 2015.

[112] Sylvio Rüdian, Niels Pinkwart, and Zhi Liu. I know who you are: Deanonymization using facebook likes. In *Workshops der INFORMATIK 2018-Architekturen, Prozesse, Sicherheit und Nachhaltigkeit*. Köllen Druck+ Verlag GmbH, 2018.

[113] Arvind Narayanan and Vitaly Shmatikov. De-anonymizing social networks. In *IEEE symposium on security and privacy*. IEEE, 2009.

[114] Meiqi Wang, Qingfeng Tan, Xuebin Wang, and Jinqiao Shi. De-anonymizing social networks user via profile similarity. In *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*. IEEE, 2018.

[115] Katchaguy Areekijseree, Ricky Laishram, and Sucheta Soundarajan. Max-node sampling: An expansion-densification algorithm for data collection. In *IEEE International Conference on Big Data*, 2016.

[116] Katchaguy Areekijseree, Ricky Laishram, and Sucheta Soundarajan. Guidelines for online network crawling: A study of data collection approaches and network properties. In *Proceedings of the 10th ACM Conference on Web Science*, 2018.

[117] Manish Kumar, Rajesh Bhatia, and Dhavleesh Rattan. A survey of web crawlers for information retrieval. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 7(6), 2017.

[118] Yilun Shang. False positive and false negative effects on network attacks. *Journal of Statistical Physics*, 170(1), 2018.

[119] Quan Bai, Gang Xiong, Yong Zhao, and Longtao He. Analysis and detection of bogus behavior in web crawler measurement. *Procedia Computer Science*, 31, 2014.

[120] Mihai Valentin Avram, Shubhanshu Mishra, Nikolaus Nova Parulian, and Jana Diesner. Adversarial perturbations to manipulate the perception of power and influence in networks. In *2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 2019.

[121] Mainack Mondal, Bimal Viswanath, Allen Clement, Peter Druschel, Krishna P Gummadi, Alan Mislove, and Ansley Post. Defending against large-scale crawls in online social networks. In *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, 2012.

[122] Paul W Olsen, Alan G Labouseur, and Jeong-Hyon Hwang. Efficient top-k closeness centrality search. In *2014 IEEE 30th International Conference on Data Engineering*. IEEE, 2014.

[123] Kazuya Okamoto, Wei Chen, and Xiang-Yang Li. Ranking of closeness centrality for large-scale social networks. In *International workshop on frontiers in algorithmics*. Springer, 2008.

[124] Michele Borassi, Pierluigi Crescenzi, and Andrea Marino. Fast and simple computation of top-k closeness centralities. *arXiv preprint arXiv:1507.01490*, 2015.

[125] Patrick Bisenius, Elisabetta Bergamin, Eugenio Angriman, and Henning Meyerhenke. Computing top-k closeness centrality in fully-dynamic graphs. In *2018 Proceedings of the Twentieth Workshop on Algorithm Engineering and Experiments (ALENEX)*. SIAM, 2018.

[126] Naveen Gupta, Anurag Singh, and Hocine Cherifi. Community-based immunization strategies for epidemic control. In *2015 7th international conference on communication systems and networks (COMSNETS)*. IEEE, 2015.

[127] Pegah Hozhabrierdi, Raymond Zhu, Maduakolam Onyewu, and Sucheta Soundarajan. Network-based analysis of early pandemic mitigation strategies: Solutions, and future directions. *Northeast Journal of Complex Systems (NEJCS)*, 3(1), 2021.

[128] Dale Weston, Katharina Hauck, and Richard Amlôt. Infection prevention behaviour and infectious disease modelling: a review of the literature and recommendations for the future. *BMC public health*, 18(1):336, 2018.

[129] Benjamin J Cowling, Diane MW Ng, Dennis KM Ip, Quiyan Liao, Wendy WT Lam, Joseph T Wu, Joseph TF Lau, Sian M Griffiths, and Richard Fielding. Community psychological and behavioral responses through the first wave of the 2009 influenza a (h1n1) pandemic in hong kong. *The Journal of infectious diseases*, 202(6):867–876, 2010.

[130] Tegan Cruwys, Mark Stevens, and Katharine H Greenaway. A social identity perspective on covid-19: Health risk is affected by shared group membership. *British Journal of Social Psychology*, 2020.

[131] John Cannarella and Joshua A Spechler. Epidemiological modeling of online social network dynamics. *arXiv preprint arXiv:1401.4208*, 2014.

[132] Theresa Kuchler, Dominic Russel, and Johannes Stroebel. The geographic spread of covid-19 correlates with the structure of social networks as measured by facebook. Technical report, National Bureau of Economic Research, 2020.

[133] Sijuan Ma, Ling Feng, and Choy-Heng Lai. Mechanistic modelling of viral spreading on empirical social network and popularity prediction. *Scientific reports*, 8(1):1–10, 2018.

[134] Petter Holme. Model versions and fast algorithms for network epidemiology. *arXiv preprint arXiv:1403.1011*, 2014.

[135] Tom Britton. Stochastic epidemic models: a survey. *Mathematical biosciences*, 225(1):24–35, 2010.

[136] Cleo Anastassopoulou, Lucia Russo, Athanasios Tsakris, and Constantinos Siettos. Data-based analysis, modelling and forecasting of the covid-19 outbreak. *PloS one*, 15(3):e0230405, 2020.

[137] Ebenezer Bonyah and Kazeem Oare Okosun. Mathematical modeling of zika virus. *Asian Pacific Journal of Tropical Disease*, 6(9):673–679, 2016.

[138] Vittoria Colizza, Alain Barrat, Marc Barthelemy, Alain-Jacques Valleron, and Alessandro Vespignani. Modeling the worldwide spread of pandemic influenza: baseline case and containment interventions. *PLoS Med*, 4(1):e13, 2007.

[139] Benjamin Ivorra, Miriam Ruiz Ferrández, M Vela-Pérez, and AM Ramos. Mathematical modeling of the spread of the coronavirus disease 2019 (covid-19) taking into account the undetected infections. the case of china. *Communications in nonlinear science and numerical simulation*, page 105303, 2020.

[140] Yuri Bruinen de Bruin, Anne-Sophie Lequarre, Josephine McCourt, Peter Clevestig, Filippo Pigazzani, Maryam Zare Jeddi, Claudio Colosio, and Margarida Goulart. Initial impacts of global risk mitigation measures taken during the combatting of the covid-19 pandemic. *Safety Science*, page 104773, 2020.

[141] Martin I Meltzer, Nancy J Cox, and Keiji Fukuda. The economic impact of pandemic influenza in the united states: priorities for intervention. *Emerging infectious diseases*, 5(5):659, 1999.

[142] Andrew Atkeson. What will be the economic impact of covid-19 in the us? rough estimates of disease scenarios. Technical report, National Bureau of Economic Research, 2020.

[143] Badar Nadeem Ashraf. Economic impact of government interventions during the covid-19 pandemic: International evidence from financial markets. *Journal of Behavioral and Experimental Finance*, 27:100371, 2020.

[144] Giovanni Bonaccorsi, Francesco Pierri, Matteo Cinelli, Andrea Flori, Alessandro Galeazzi, Francesco Porcelli, Ana Lucia Schmidt, Carlo Michele Valensise, Antonio Scala, Walter Quattrociocchi, et al. Economic and social consequences of human mobility restrictions under covid-19. *Proceedings of the National Academy of Sciences*, 117(27):15530–15535, 2020.

[145] Calistus N Ngonghala, Enahoro A Iboi, and Abba B Gumel. Could masks curtail the post-lockdown resurgence of covid-19 in the us? *Mathematical biosciences*, 329:108452, 2020.

[146] Harald Brüssow. Covid-19: Test, trace and isolate-new epidemiological data. *Environmental Microbiology*, 2020.

[147] Royal Society DELVE Initiative. Test, trace, isolate. http://web.archive.org/web/20080207010024/http://www.808multimedia.com/winnt/kernel.htm. Accessed: 2020-08-21.

[148] Aram Galstyan and Paul Cohen. Cascading dynamics in modular networks. *Physical Review E*, 75(3):036109, 2007.

[149] Alexandru Topîrceanu. Analyzing the impact of geo-spatial organization of real-world communities on epidemic spreading dynamics. In *International Conference on Complex Networks and Their Applications*, pages 345–356. Springer, 2020.

[150] Jérôme Kunegis. Konect: the koblenz network collection. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 1343–1350, 2013.

[151] Clara Stegehuis, Remco Van Der Hofstad, and Johan SH Van Leeuwaarden. Epidemic spreading on complex networks with community structures. *Scientific reports*, 6(1):1–7, 2016.

[152] Peiyan Yuan and Shaojie Tang. Community-based immunization in opportunistic social networks. *Physica A: Statistical Mechanics and its Applications*, 420:85–97, 2015.

[153] Yasuko Matsubara, Yasushi Sakurai, B Aditya Prakash, Lei Li, and Christos Faloutsos. Rise and fall patterns of information diffusion: model and implications. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 6–14, 2012.

[154] Petter Holme. Efficient local strategies for vaccination and network attack. *EPL (Europhysics Letters)*, 68(6):908, 2004.

[155] Lorenzo Pellis, Frank Ball, Shweta Bansal, Ken Eames, Thomas House, Valerie Isham, and Pieter Trapman. Eight challenges for network epidemic models. *Epidemics*, 10:58–62, 2015.

[156] Stanley Wasserman, Katherine Faust, et al. Social network analysis: Methods and applications. 1994.

[157] Per Block, Marion Hoffman, Isabel J Raabe, Jennifer Beam Dowd, Charles Rahal, Ridhi Kashyap, and Melinda C Mills. Social network-based distancing strategies to flatten the covid-19 curve in a post-lockdown world. *Nature Human Behaviour*, 4(6):588–596, 2020.

[158] Martin I Meltzer, Nancy J Cox, and Keiji Fukuda. The economic impact of pandemic influenza in the united states: priorities for intervention. *Emerging infectious diseases*, 5(5):659, 1999.

[159] Reuven Cohen, Shlomo Havlin, and Daniel Ben-Avraham. Efficient immunization strategies for computer networks and populations. *Physical review letters*, 91(24):247901, 2003.

[160] Ronald John Johnston. Social distance, proximity and social contact: Eleven cul-de-sacs in christchurch, new zealand. *Geografiska Annaler: Series B, Human Geography*, 56(2):57–67, 1974.

[161] Jacob Goldenberg and Moshe Levy. Distance is not dead: Social interaction and geographical distance in the internet era. *arXiv preprint arXiv:0906.3202*, 2009.

[162] Shenghao Yang, Priyabrata Senapati, Di Wang, Chris T Bauch, and Kimon Fountoulakis. Targeted pandemic containment through identifying local contact network bottlenecks. *arXiv preprint arXiv:2006.06939*, 2020.

[163] Andreas Radbruch and Hyun Dong Chang. A long-term perspective on immunity to covid. *Nature News and Views*, 2021.

[164] Paul S Bradley, Kristin P Bennett, and Ayhan Demiriz. Constrained k-means clustering. *Microsoft Research, Redmond*, 20(0):0, 2000.

[165] P Sapiezynski, A Stopczynski, DD Lassen, and SL Jørgensen. The copenhagen networks study interaction data. figshare, 2019.

[166] Chuanli Huang, Min Wang, Warda Rafaqat, Salman Shabbir, Liping Lian, Jun Zhang, Siuming Lo, and Weiguo Song. Data-driven test strategy for covid-19 using machine learning: A study in lahore, pakistan. *Socio-economic planning sciences*, 80:101091, 2022.

[167] Ariel Félix Gualtieri, Carolina de la Cal, Augusto Francisco Toma, and Pedro Hecht. Spread of sars-cov-2 in a sis model with vaccination and breakthrough infection. *arXiv preprint arXiv:2206.01803*, 2022.

[168] Matthieu Nadini, Kaiyuan Sun, Enrico Ubaldi, Michele Starnini, Alessandro Rizzo, and Nicola Perra. Epidemic spreading in modular time-varying networks. *Scientific reports*, 8(1):1–11, 2018.

Vita

Education

- Master of Science in Computer Science,
  Syracuse University, Syracuse, NY,
  Graduation: May 2020 (GPA: 3.9/4).
- Bachelor of Science in Electrical Engineering & Telecommunications,
  K. N. Toosi University of Technology, Tehran, Iran,
  Graduation: May 2016 (GPA: 18.14/20).

Experience

- Research Assistant, Syracuse University (2016 – Present).
- Visiting Researcher, University of Barcelona (2021 – 2022).
- Data Analytics Intern, Thales UTM (2019 – 2020).
- Teaching Assistant, Syracuse University (2016 – 2019).

Publications

- **[Conference Paper] Hozhabrierdi, Pegah**, and Sucheta Soundarajan "*CoVerD: Community-based Vertex Defense against Crawling Adversaries*", International Conference on Complex Networks and Their Applications. Springer, 2021.

- **[Conference Paper]** Laishram, Ricky, **Pegah Hozhabrierdi**, Jeremy Wendt, and Sucheta Soundarajan. "*NetProtect: Network Perturbations to Protect Nodes against Entry-Point Attack*", ACM Web Science Conference (WebSci), 2021.

- **[Journal Paper] Hozhabrierdi, Pegah**, Raymond Zhu, Maduakolam Onyewu, and Sucheta Soundarajan. "*Network-Based Analysis of Early Pandemic Mitigation Strategies: Solutions, and Future Directions*", Northeast Journal of Complex Systems (NEJCS) 2021.

- **[Conference Paper] Hozhabrierdi, Pegah**, Dunai Fuentes Hitos, and Chilukuri K. Mohan. "*Zero-Shot Source Code Author Identification: A Lexicon and Layout Independent Approach*", International Joint Conference on Neural Networks (IJCNN). IEEE 2020.

- **[Conference Paper] Hozhabrierdi, Pegah**, and Reza Zafarani, "*The Impact of Graph Structure on Small-World Shortest Paths*", International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation (SBP-BRiMs). Springer, 2019.

- **[Conference Paper] Hozhabrierdi, Pegah**, Dunai Fuentes Hitos, and Chilukuri K. Mohan, "*Python Source Code De-Anonymization Using Nested Bigrams*", International Conference on Data Mining Workshops (ICDMW). IEEE, 2018.

Under Review

- **[Conference Paper] Hozhabrierdi, Pegah**, and Sucheta Soundarajan. "*ComMit: Blind Community-based Early Mitigation Strategy against Viral Spread*"

- **[Journal Paper]** Laishram, Ricky, **Pegah Hozhabrierdi**, Jeremy Wendt, and Sucheta Soundarajan. "*On Efficiently Hiding Nodes from Entry-Point Attacks*"

Extended Abstract

- **[Conference Oral Presentation] Hozhabrierdi, Pegah**, and Sucheta Soundarajan "*Fast and Efficient Node Protection against Crawling Attacks*", International School and Conference on Network Science (NetSciX), 2022.

Honors & Awards

- Syracuse University Research Excellence Doctoral Funding Fellowship (Spring 2022).
- Young Researchers of the Complex System Society Scholarship (Winter 2022).
- RESORC Honorary Speaker Award (Winter 2022).
- Syracuse University GSO Travel Grant (Fall 2021).
- International Society for Music Information Retrieval Grant (Fall 2021).
- ICML Workshop on Computational Biology Fellowship (Summer 2021).
- SBP-BRiMS Conference Travel Award (2019 & 2020).
- WCCI Congress Travel Award (Summer 2020).
- NeurIPS Conference Travel Award (Fall 2019).
- WiML Workshop Travel Award (Fall 2019).
- ICML Travel Award (Summer 2019).
- Syracuse University Research Day Best Poster Award (Spring 2019).
- Syracuse CASE Center Research Assistantship (Summer 2018).
- Syracuse University Graduate Fellowship (2016 – 2018).
- German Academic Exchange Service (DAAD) Scholarship (Summer 2015).
- Jade University Summer Program Scholarship (Summer 2014).

Academic Services

- **Senior PC Member**: ICML 2022.
- **PC Member**: AAAI 2022, NeurIPS 2022, ICLR 2021/2022, ICML 2020/2021, NeurIPS WiML Workshop 2019.

- **External Reviewer**: WebConf 2019/2020, AAAI 2018/2019, ICCS 2019, WSDM 2018, KDD 2018/2019/2020, TKDD 2018, ICDM 2020/2021/2022, SBP-BRiMS 2017/2018/2019.
- **Volunteer**: NeurIPS 2021/2022, ICLR 2021, ICML 2021.

Extracurricular & Certificates

- Winter Workshop on Complex Systems, Besançon, France (Winter 2022).
- AccelNet-MultiNet Fellow Program – Funded by NSF (2021 – 2023).
- Introduction to Genetics & Evolution, Duke University MOOC (January 2022).
- Scientific Data Analysis at Scale (SciDAS) Cloud Computing Workshop, University of Wisconsin-Madison (December 2021).
- Machine Learning in Bioinformatics Summer School, HSE University (August 2021).
- Eastern European Machine Learning (EEML) Summer School (August 2020).
- Oxford Machine Learning (OxML) Summer School, University of Oxford (July 2020).