Data-driven Studies on Social Networks: Privacy and Simulation

by

Yasanka Sameera Horawalavithana

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Computer Science and Engineering
College of Engineering
University of South Florida

Major Professor: Adriana Iamnitchi, Ph.D.
John Skvoretz, Ph.D.
Lawrence O. Hall, Ph.D.
Giovanni L. Ciampaglia, Ph.D.
Michael Maness, Ph.D.

Date of Approval:
June 16, 2021

Keywords: Graphs, Machine Learning, Anonymization, Information Diffusion, Twitter

Dedication

This dissertation is dedicated to my parents, my wife and my lovely son. I am grateful to have you in my life.

Acknowledgments

First, I want to thank my advisor Adriana Iamnitchi. She was a fantastic mentor, and most importantly a very good friend. I was fortunate to work with her for last five years where she inspired me to be a good researcher.

I thank my defense committee members, John Skvoretz, Lawrence O. Hall, Giovanni L. Ciampaglia, and Michael Maness for the feedback received to improve this dissertation.

I had awesome collaborators. A few of them are Clayton Gandy, Juan Arroyo Flores, Essa Alhazmi, Kin Wai NG, Nazim Choudhury, Abhishek Bhattacharjee, and Renhao Liu.

I am very fortunate to have good friends, some are my roommates, Chatura Wickramaratne, Sachin Wickramaarachchi, Nalaka Kapuruge, and others are my lab mates Sathyanarayanan Aakur, Subramanian Viswanathan, and Sreeja Nair.

This work was supported by the DARPA SocialSim Program and the Air Force Research Laboratory under contract FA8650-18-C-7825, and National Science Foundation (NSF) in USA under the grant IIS 1546453. I also thank Pacific National Northwest Laboratory (PNNL), and Leidos for their support through out the SocialSim project for providing data and evaluation code.

My parents, my grandmother, my sister, my aunt, and my uncle always give me the courage and strength to continue my studies.

Lastly, I am truly blessed to have my wife, Keshani who supported me throughout my doctoral studies, especially during the pandemic where we had our first son, Keyal.

Table of Contents

List of Tables

List of Figures

Abstract

Social media datasets are fundamental to understanding a variety of phenomena, such as epidemics [1], adoption of behavior [2], crowd management [3], and political uprisings [4]. At the same time, many such datasets capturing computer-mediated social interactions are recorded nowadays by individual researchers or by organizations. However, while the need for real social graphs and the supply of such datasets are well established, the flow of data from data owners to researchers is significantly hampered by privacy risks: even when humans' identities are removed, or data is anonymized to some extent, studies have proven repeatedly that re-identifying anonymized user identities (i.e., de-anonymization) is doable with high success rate [5, 6, 7, 8].

A main research challenge is to develop a principled understanding of how to measure the effectiveness of an anonymization scheme and thus, conversely, the likely success of a de-anonymization attack [9]. This dissertation develops methods to understand what makes some graph datasets more resilient to de-anonymization attacks. We propose a data-driven framework to 1) quantify the vulnerability of a graph to a re-identification attack; 2) quantitatively identify which graph structural properties contribute most to graph vulnerability; and 3) propose guidelines to develop new methodologies related to graph anonymization, de-anonymization and graph vulnerability quantification. We show the usefulness of this framework on a large set of synthetically generated graphs with controlled propertied inspired from a set of real social networks. Thus, we provide an unified framework to analyze the privacy/utility trade-off imposed on any family of social graphs.

We extend this data-driven framework for networks with node attributes. Using this improved framework, we quantify how much better a node re-identification attack performs when the node attributes are included in the attack compared to when there is no node attribute information available to the attacker. We quantify the privacy impact of node attributes under an attribute attachment model biased towards homophily, and analyze the interplay between graph structures and attribute information. Our results show that binary node attributes increase the chance of revealing node identity independent of their placements in the network. Further, we show that other network properties independent of the degree distribution put node privacy at risk. This improves the current understanding of graph privacy, as it means that protecting graph privacy is much harder than previously considered [10, 11].

Once privacy is guaranteed to a certain level, social media datasets are useful for various studies. One such important study is to analyze and model the information spreading patterns on social networks. Understanding how information (e.g., opinions, rumours, etc.) spreads on social networks has many benefits ranging from controlling the spread of bad rumour [12], identifying influential spreaders [13], reducing the harm of an outbreak, etc. [14]. Although there are a variety of classical diffusion models developed for epidemic spreading [15], they are not representative for capturing the information spread in social media. This dissertation contributes to the development of data-driven models to predict social media activity.

In this line of work, we first develop methods to forecast how conversations will evolve on a social media platform. Given a set of original posts on a social platform, such as posts on Reddit in a continuous interval of time, we predict the conversation trees rooted in these seeds. For each conversation, we predict the final shape of the message tree, the user who posts each message, and

the time (in continuous space) of the posting of each message. Our solution uses a probabilistic generative model with the support of a genetic algorithm and Long-Short Term Memory (LSTM) neural networks. We evaluate the proposed approach on real world conversations as appeared on subreddits related to crypto-currency and cyber-security on Reddit. We show that this technique can generate accurate conversation topological structures over time, and can accurately predict the volume of messages and the engagement of users over time.

We improve this technique to predict the Twitter activities per topic of interest during a political crisis period. By their nature, periods of crisis do not include many repeatable events, thus it is difficult to learn and predict how social media users react. We use external events information as seen through the lens of physical conflict and news when improving the simulator design. Specifically, we use the time-aligned exogenous signals to predict when tweets are posted, in which topic, and by which user. We use the previously developed cascade generation model to predict the resharing activity. We evaluate this finer-granularity of simulations by the volume and temporal pattern of Twitter discussions, new user engagements and the structure of user interaction network. We show on Twitter data collected during the Venezuela political crisis that our model generates activities that follow the ground truth.

Chapter 1: Introduction

Social media platforms such as Twitter, Reddit, YouTube, and Facebook are popular over the last few years because they offer useful services for people to connect and interact with each other. These platforms offer large network datasets that often represent social interactions between real-world entities like friendship, follower, and professional relations. These datasets are helpful for a variety of research studies such as community evolution [16], opinion polarization [17], disaster response [18], racial/ethnic disparities [19], stress detection [20], etc.

This dissertation focuses on two important studies of social media: protecting privacy of individuals in publicly available social network data, and simulating online user activity in various social media platforms. The first study was motivated by the access and privacy issues of social network data due to the sensitive information they capture. For example, there are serious privacy issues raised when social network data leaks political leanings, sexual preferences, corporate credentials, etc. [21]. The second study aims to accurately model information dissemination in social media across various contexts. Being able to forecast social media activity in the future has immediate applications. For example, platform curators can predict users who may post inflammatory messages in a conversation, and monitor/censor their activity. Other benefits include the evaluation of intervention strategies to limit disinformation.

1.1 Privacy of Social Network Data

Social networks have substantial scientific value to the research community but public release of such data may jeopardize the privacy of individuals. There are numerous ways that privacy can be breached. For example, an adversary might be interested to find out whether a particular user is active on a certain political forum, or whether there is a relationship between two users in a dating network, or whether a group of users in a neighborhood voted for a particular candidate. A number of data protection methods have been proposed to mitigate the privacy invasion of individuals [22]. For example, a user's identity may be protected via naive sanitizing, by simply removing the identifiable attributes from the publicly available data, or by structural anonymization, in which nodes and edges in the social network are removed/inserted to obscure the original topological structure. However, data breaches happen regularly where adversaries use sophisticated techniques to defeat data protection mechanisms. The de-anonymization attack on *Data for Development* (D4D) challenge data [23] is a good example on breaching the privacy of individuals from poorly sanitized public data. The D4D datasets represent "anonymized" call records and SMS exchanges that were extracted from the users of major communication network in the Ivory Coast. Yet the adversaries revealed the identity of users using a powerful de-anonymization attack [24]. They used the information from different anonymized subgraphs to decode the anonymized user accounts.

An important question is how to effectively anonymize graphs without destroying their utility [25]. For example, preserving particular network characteristics (e.g., degree distribution, clustering, etc.) in the anonymized graph is important for the end application. Typically, to the extent these methods preserve utility, the anonymized graphs are vulnerable to modern de-anonymization

atacks [9]. What is not well understood, however, is the interplay between the anonymity guarantees that these anonymization methods provide vs. the strength of the attack and the particularities of the dataset to be anonymized. Or, whether some networks are inherently more "anonymizable," that is, immune to strong attacks even using weak structural anonymization schemes. A main research challenge is to develop a principled understanding of how to measure the effectiveness of an anonymization scheme and thus, conversely, the likely success of a de-anonymization attack. In this dissertation, we try to understand what makes some graph datasets more resilient to de-anonymization attacks.

1.2 Simulating Social Media Activity

Understanding how information is disseminated in online social environments has significant real-world impact, from health care to marketing. Significant effort has been invested in characterizing information diffusion in various platforms. For example, Cheng et al. [26] characterized the types of information cascades in Facebook. They showed the types of cascades depend on the factors related to the effort and social cost of user participation. Zuo et al. [27] studied the social contagion of cheating behavior in online gaming platforms. Vosoughi et al. [28] determined based on a collection of tweets of political news that false information spreads faster, farther, deeper and broader than true facts. This phenomenon may be explained by human factors such as emotional reaction to surprise, fear and disgust that are more likely induced by fabricated news.

Our goal is to develop a social simulator that captures the information dissemination within and across various social media platforms. A simulator is more useful when it is able to predict realistic online user activities at fine granularity (who responds to whom on which topic, and when)

in a future time horizon without having the ground truth activity. Although simple to state, this granularity of predictions is shown [29] to be difficult to make in part because of the irregular patterns of information flows due to the influence of both internal and external factors, and in part because different social platforms have different algorithms for content promotion. A reliable simulator can realistically respond to internal and external stimuli by: 1) capturing peaks of activity on particular subjects of interest; 2) responding realistically to the timing of external events and internal amplified discussions; 3) capturing activity per topic, where topics can be loosely related; and 4) representing accurately the size of the newly engaged audience, that can vary significantly over time and with topics.

Simulating user activities in online social media platforms has many benefits. These predictions can be used to study "what if" scenarios in an operational setup. For example, what response would be generated if a particular post is made by a particular user account? That is, how large of a reaction would that generate in terms of messages and user engagement over time? What if that same message is posted by a different user? (say, a government organization vs. a bot account?). On the other hand, researchers could test the effects of intervening within the platform to influence activity: would the blocking of some accounts significantly impact a disinformation campaign? How late in an information operation would an intervention be effective, knowing that it may take some time to identify the information campaign and its operators? Other applications for such a simulator include generating realistic datasets for filling in gaps in data collected for various scientific enquiries; studying cross platform information diffusion; or identifying users who aim to promote violence during an election season.

1.3 Contributions and Outline

We make multiple contributions in this dissertation towards developing data-driven models using social network data.

- Chapter 2 introduces a data-driven framework to measure privacy and utility on network data. We develop methods to examine the interplay between graph properties and the vulnerability to de-anonymization attacks. We demonstrate its applicability via extensive experiments on thousands of graphs with controlled properties generated from real datasets. In addition, we show empirically that there are structural properties that affect graph vulnerability to re-identification attacks independent of degree distribution.

- Chapter 3 extends this framework to explore the interplay between graph topology and attribute placement with respect to the anonymity. We quantitatively study the impact of binary node attributes on node privacy. Our experiments show that the population's diversity on the binary attributes consistently degrades anonymity. The content of these two chapters is primarily based on our published work [30, 31, 32].

- Chapter 4 introduces the related problems of simulating social media activity. We describe the challenges in this problem space, discuss the related attempts, and explain the problem scenarios that motivate the design of social simulators developed as a part of this dissertation.

- Chapter 5 proposes a data-driven method that forecasts groups of topic-related, overlapping, online conversation trees on Reddit. Our method is generative: given a group of original posts, it generates the resulting conversation threads with timing and authorship information. We demonstrate using two large datasets from Reddit that the microscopic properties of

such groups of conversations can be accurately predicted when starting from the original posts, without knowledge of the intermediate reactions to such posts. We show that our solution significantly outperforms competitive baselines in terms of predicting the conversation structure and user engagement over time.

- Chapter 6 presents the design, implementation and evaluation of a simulator that generates Twitter activity related to a political crisis using signals from contemporary exogenous data, such as news articles and Reddit. The simulator is composed of multiple modules, each specialized to accurately predict a dimension of the activity, such as the number of tweets, or the retweet cascades. We use the cascade solution presented in Chapter 5 to predict the growth of retweet cascades, thus testing its generality across two platforms, Reddit and Twitter. Most importantly, the simulator generates activity as it pertains to a particular topic from the overall conversation of interest. We use the Venezuela political crisis from the beginning of 2019 as the scenario on which we train and test the simulator. We describe our experience on building this simulator, including the failed attempts at capturing peaks and lows in social media activity. The content of Chapter 4-6 is primarily based on experience from DARPA SocialSim Challenges.

- Chapter 7 concludes with a discussion of our contributions and the future work.

Chapter 2: Privacy of Network Topology[1]

Social networks are often mined to uncover insights about the structure and function of the interactions represented. This substantial scientific value to the research community comes with risks: the release of such data may jeopardize the privacy of individuals.

The AOL [33] and Netflix [34] scandals are textbook examples on breaching the privacy of individuals by publicly releasing poorly sanitized data. The first scandal was related to the public release of anonymized search logs by AOL in 2006 [35]. These records contained web search queries of more than 500,000 Americans who used the AOL search engine for three months. Two New York Times journalists matched the personally identifiable information present in these anonymized records with the publicly available phone book listings to decode a few user identities. The most popular re-identified account was the user No. 4417749, Thelma Arnold, a 62-year-old widow who searched for topics such as "numb fingers", "60 single men" and "dog that urinates on everything" [35]. It revealed that many other user accounts ranging from cancer patients, pregnant mothers to college students were also re-identifiable using a similar methodology. This privacy violation led to a class action lawsuit against AOL at the end [36]. The second scandal was related to the public release of Netflix movie ratings as a part of Netflix movie recommendation challenge [37]. Two academic researchers matched these records with the Internet Movie Database (IMDb) ratings [34]. They were able to identify many users present in both datasets even though their identities were anonymized in the Netflix dataset.

[1]This chapter was previously published in [30]. Permission is included in Appendix A.

Many anonymization methods have been proposed to mitigate the privacy invasion of individuals from the public release of graph data [22]. The accepted approach now is to anonymize social graphs by modifying the graph structure enough to decouple the particular node identity from its social ties, yet preserving the graph characteristics in aggregate. Various solutions have been proposed, some based on rewiring the original graph structure, others based on clustering, and others based on generating graphs from a graph signature. For all structural graph anonymization techniques, however, the challenge is the tension between providing privacy in the altered graph structure and preserving the accuracy of the structural characteristics of the original graph in the altered graph, which is what matters for their utility for research [38]. In this method, the anonymized graph is isomorphic to the original preserving the structural data utility which in turn makes it the most vulnerable instance to basic de-anonymization attacks [39]. At the other extreme, the generation of random graphs could be considered as an anonymization method to generate a complete non-isomorphic graph to the original. Though this method achieves a higher level of privacy, significant loss of original graph structure may affect the fidelity of anonymized data usage. Typically, to the extent many anonymization methods preserve utility, the anonymized graphs are vulnerable to modern de-anonymization attacks [9]. Thus, an important question is how to effectively anonymize graphs without destroying their utility while protecting the privacy of users [25].

Various studies touched on this problem, typically in the context of specific anonymization techniques and specific desired utility metrics [40, 9]. For example, Ji et. al. [40] present a benchmark study on comparing perturbation-based anonymization schemes with respect to the preserved utility and the resistance to specific de-anonymization attacks. Missing from the state of the art is a systematic understanding of the limitations on anonymity that utility objectives impose. Specifi-

cally, we ask: *Which graph properties give away most information such that a large fraction of the nodes can be identified*? Understanding the answer to this question is beneficial in many practical ways. First, it can help the data practitioner in deciding which graph properties *should not* be preserved in the anonymized version of the dataset, in an attempt to increase node anonymity. For example, if the joint degree distribution is shown to be revealing too much information (as it was, in fact, shown in [41]), then an anonymization technique that preserves the degree distribution of the original graph dataset should be understood that it comes with significant risks in terms of privacy and may be avoided. Second, new anonymization techniques may be designed with the specific objective of obscuring in the anonymized graph the very properties of the original graph that proved to be too revealing. Thus, for example, if for a particular network the degree assortativity (defined as the tendency of nodes with similar degrees to be connected by an edge) significantly helps in node re-identification, then an anonymization algorithm that perturbs the assortativity coefficient may be needed. This observation opens a new path in the space of graph anonymization techniques, where the typical design objectives include the preservation of some structural properties, rather than their explicit perturbation.

In this chapter, we propose a modeling framework to 1) quantify the vulnerability of a graph to a re-identification attack; and 2) quantitatively identify which structural properties contribute most to graph vulnerability. We show the usefulness of this framework on a large set of synthetically generated graphs with controlled properties inspired from a set of real social networks.

This chapter makes the following contributions. First, we introduce a new question which, while related to previously asked questions, opens a new research direction. Specifically, we ask: *which network characteristics make a graph more vulnerable to a de-anonymization attack?* Answer-

ing this question can guide data practitioners to navigate among many anonymization techniques and utility requirements. Second, we propose a framework [42] that answers empirically this question. Third, as a proof of concept, we instantiate this framework by employing a representative set of network metrics, a strong machine learning based de-anonymization attack, and thousands of graphs with controlled characteristics. And fourth, our experiments show how several graph metrics have a combined effect on graph vulnerability under the de-anonymization attack considered.

The rest of the chapter is structured as follows. Section 2.1 presents the related work. Section 2.2 introduces the framework and our proof-of-concept instantiation of its modules. Section 2.3 presents the real networks and the families of synthetic datasets used in our empirical study. Section 2.4 analyzes the relationships between graph properties and vulnerability to node re-identification. And finally, Section 2.5 concludes with discussions of our contributions.

2.1 Related Work

Much progress has been made in the last decade on problems related to graph anonymization. To place our results in the vast literature on graph anonymization, we discuss related work structured around our main contributions.

2.1.1 Graph Privacy and Utility

Because utility is typically expressed as (distance between) graph metrics and graph metrics describe network properties, our question of which network properties makes graphs vulnerable to de-anonymization attacks is closely related to the question of utility vs. anonymity. Significant effort has been invested in understanding the inherent tension between achieved privacy and preserved utility on publishing graph datasets [41, 22]. For example, while any anonymization scheme that

preserves the degree distribution is vulnerable to de-anonymization attacks [9, 40], perturbations to the degree distribution in the anonymization process lead to significant utility loss, that is, to perturbations of important graph properties in the anonymized graphs [43]. The fraction of nodes with only one neighbor is an important factor in maintaining anonymity: intuitively, they carry little information to reveal the identity of their (only) neighbor [9]. Moreover, it has been shown experimentally that utility is degraded faster than privacy is achieved [44, 38].

Theoretical frameworks were proposed to quantify the tradeoff between privacy and utility. Ji et al. [45] introduced a theoretical model to quantify the de-anonymizablity of graph datasets by considering the topological importance of nodes. They inferred that privacy is affected by high average degree. Lee et al. [46] analyzed the relation between the utility of an anonymized graph and its vulnerability to a common neighbor-based node re-identification attack. They formulate conditions for the success of de-anonymization attacks based on two distance-based utility metrics between the anonymized (or auxiliary) and the original graph.

The differences between the privacy vs. anonymity investigations and our focus are the following: First, our question addresses the original graph properties rather than the anonymized version. Thus, answers to this question are independent of any anonymization techniques, but instead apply to the intrinsic properties of the original network. Second, by not focusing on utility we are not restricted to selecting a subset of "useful" properties of the network for a particular context, and thus our question allows for a wider investigation of graph properties and their effect.

2.1.2 De-anonymization Attack Models and Success Metrics

A well accepted graph de-anonymization attack uses information from an auxiliary graph in order to re-identify the nodes in an anonymized graph [47]. The success of such an attack is determined by the rate of correct re-identification of the original nodes in the network. In general, de-anonymization attacks harness structural characteristics of nodes that are uniquely distinguishable [22]. Many such attacks can be categorized into *seed-based* and *seed-free*, based on the prior seed knowledge available to an attacker [22].

In seed-based attacks, the process of de-anonymization is conducted to re-identify nodes and ties with the support of sybil nodes [48] or some known mappings of nodes in an auxiliary graph [5, 6, 7, 49, 8]. The effectiveness of such attacks is influenced by the quality of the seeds [9].

In seed-free attacks, the problem of de-anonymization is usually modeled as a graph matching problem [50] (also known as the network alignment problem [51]). On aligning networks, the goal is to find the correct mapping between the node sets of two structurally correlated graphs. Recent work suggests information-theoretic conditions when this perfect mapping is possible [52, 53, 51, 54]. Most of these studies are based on Erdös-Rènyi models (theoretical models without representation in real datasets) and assume unlimited computational resources, while others make impractical assumptions about the seed knowledge, such as the availability of hub nodes as seeds [55].

Several research efforts have proposed statistical models for the re-identification of nodes without relying on seeds, such as the Bayesian model [50] or optimization models [56, 39]. Many heuristics were taken into account for the propagation process of re-identification, exploiting graph characteristics such as degree [57], k-hop neighborhood [58], linkage-covariance [38], eccentricity [47], or community [59].

Some anonymization techniques rely on perturbing a set of edges in the original graph within the limits of a given privacy budget [22]. For example, differential privacy captures the amount of noise injected [41], which is also a popular theoretical metric of quantifying the privacy of an anonymized graph. However, differential privacy is highly sensitive to the privacy budget which measures the maximum number of queries acceptable without leaking secrets [60]. Moreover, privacy metrics based on differential privacy have been shown to over-estimate privacy gains [44].

Sharad [9] proposed a general threat model to measure the success of a de-anonymization attack which is independent of the anonymization scheme. He proposed a machine learning framework to benchmark perturbation-based graph anonymization schemes. This framework explores the hidden invariants and similarities to re-identify nodes in the anonymized graphs [24]. Importantly, this framework can be easily tuned to model various types of attacks. We build on Sharad's approach in this study.

2.2 Modeling Privacy Based on Network Properties

Our main objective is to quantify the relationship between a graph's structural properties and the risk to the privacy of its nodes. We call node privacy the ability to keep the identity of a node protected. Intuitively, in a regular graph—where all nodes have the same number of neighbors—nodes are private: it is impossible, based on topological information only, to distinguish a node from the others. At the other end of the spectrum, the core of a star topology is easy to identify with some extra information. Real graph datasets lay in between these examples.

A node's identity may be protected via naive sanitization, by simply removing the identifiable attributes of the node, or by structural anonymization, in which nodes and edges in the graph are

removed/inserted to obscure the original topological structure. In this work we do not need to differentiate between these two scenarios, as the question we ask attempts to relate the properties of a graph—whether original or structurally perturbed—and the privacy of its nodes. Specifically, we ask: *Given a graph topology, which of its structural properties reveal most information that can be used to identify its nodes?* Note that if the graph of interest is the original topology of a network, then the question relates to the intrinsic vulnerability of a dataset to a re-identification attack. If the graph is already perturbed, then the question refers to the vulnerability of the structurally anonymized network to a de-anonymization attack. In this chapter, we use re-identification and de-anonymization attacks interchangeably.

2.2.1 Framework

To answer this question, we developed a framework as shown in Figure 2.1. The framework takes as input a graph dataset and contains three main components. One component, called the Attack Model in the figure, implements a re-identification attack on the input dataset and outputs a vulnerability score. Any attack algorithm can be plugged in to this component. For experimentation, we implemented a machine-learning algorithm (described in Section 2.2.2.2) based on an accepted threat model [52] (presented in Section 2.2.2.1). The definition of the vulnerability score depends on the attack model implemented.

The second component of this framework, called "Network Analysis" in the figure, performs traditional network measurements. Any metrics of interest can be output from this component in the form of numerical values or distributions. Since there are many well established tools for network analysis in the form of libraries implemented in Python, R, C++, etc., we do not need to provide any more details here.

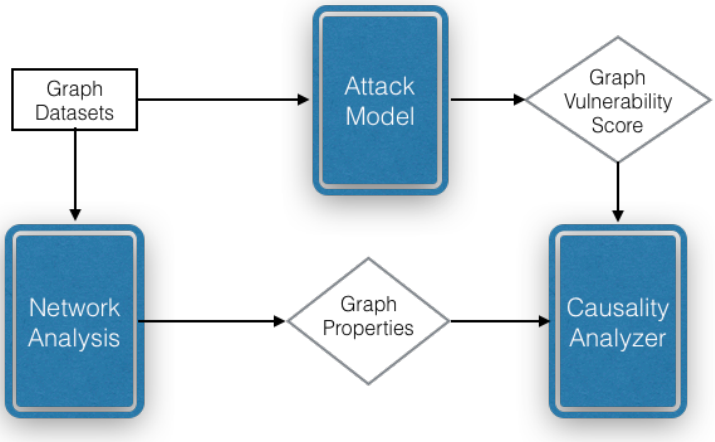


Figure 2.1: Framework to measure privacy and utility. We analyze the relationship between graph vulnerability and graph properties.

Finally, the network metrics of interest and the vulnerability score of the original graph are the input of the third component, the Causality Analyzer. This component performs a rigorous analysis of the relationship between graph vulnerability and its structural properties. The output from this component is providing a statistical answer to the question of interest.

2.2.2 The Attack Model

In order to quantify the vulnerability of a graph to node re-identification attacks, we employ a machine learning-based approach that aims at finding a bijective mapping between nodes in two different but overlapping graphs.

2.2.2.1 The Threat Model

We consider the classical threat model [52] in which the attacker aims to match nodes from two networks whose edge sets are correlated. A real-life scenario corresponding to this threat model is as follows. Let us assume there is a privacy breach over the Unix accounts of some students in a

Computer Science department: the accounts of those who accessed Facebook and Twitter from the university network are thus compromised. Consequently, an attacker has a partial view of possibly overlapping Facebook and Twitter subgraphs: some individuals are present in both graphs, even if their identities have been removed. The attacker's task is to find a bijective mapping between the two subsets of nodes in the two graphs that correspond to individuals present in both networks.

Formally, we assume that the adversary has a sanitized graph G_{san} that could be associated with an auxiliary graph G_{aux} for the re-identification attack. In the scenario discussed above, G_{san} is the Facebook network, while G_{aux} is the Twitter network of the students affected.

In order to model this scenario using real data, we split a real dataset graph $G = (V, E)$ into two subgraphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, such that $V_1 \subset V$, $V_2 \subset V$ and $V_1 \cap V_2 = V_\alpha$, where $V_\alpha \neq \phi$. The fraction of the overlap α is measured by the Jaccard coefficient of two subsets: $\alpha = \frac{|V_1 \cap V_2|}{|V_1 \cup V_2|}$. In the shared subgraph induced by the nodes in V_α, nodes will preserve their edges with nodes from V_α but might have different edges to nodes that are part of $V_1 - V_\alpha$ or part of $V_2 - V_\alpha$.

In an optimistic scenario, an attacker has access to a part of the original graph (e.g., G_1) as auxiliary data and to an unperturbed subgraph (e.g., G_2) as the sanitized data whose nodes the attacker wants to re-identify. It is also possible to split G_1 and G_2 recursively into multiple overlapping graphs, maintaining the same values of overlap parameters as above. This allows us to assess the feasibility of the de-anonymization process for large networks by significantly reducing the size of G_1 and G_2.

The resulting graphs are now the equivalent of the Facebook and Twitter networks we used as an example above. The overlap is the knowledge repository that the attacker uses for

de-anonymization [61]. Part of this knowledge will be made available to the machine learning algorithms.

Intuitively, the larger α, the more successful the attack. However, the relative success of attacks under different anonymization schemes is observed to be independent of α [9]. In order to experiment with various strengths of the attack for a fixed value of $\alpha = 0.2$, we constructed V_α in four different ways: i) as a random collection of nodes from the original graph G (R); ii) by selecting the highest degree nodes from G (HD); iii) by building a breadth-first-search tree starting from a randomly selected node in G (BFS-R); and iv) by building a breadth-first-search tree starting from the highest degree node in G (BFS-HD).

2.2.2.2 The Attack Algorithm

As previously discussed, many de-anonymization attacks can be implemented in this framework. We chose to implement the attack algorithm based on a machine learning approach for a number of reasons. First, machine learning techniques have proven successful in many real life instances of the context of graph de-anonymization [24]. Second, machine learning approaches automatically discover recognizable patterns, and thus they implicitly cover many algorithmic approaches for node re-identification. Therefore, they mount a powerful attack that can be used as benchmark for future studies.

Intuitively, a machine learning attack uses the information about the users in the two networks from the example scenario above to learn structural network patterns. It then uses these patterns to match different nodes based on similar structural characteristics. Each node is represented for learning by a set of features, as explained below.

We chose to use neighborhood degree distribution (NDD) to construct the feature vector for each node. NDD is a popular representation method due to its robustness to noise in distinguishing nodes in the graph [24, 62] and for its generality [61]. Degree-based features are also shown to be better counterparts than common-neighbor features for the performance of percolation-based de-anonymization algorithms [8].

NDD of a user u is a vector of positive integers where $NDD_u^q[k]$ represents the number of u's neighbors at distance q with degree k. We concatenate the binned version of NDD_u^1 with the binned version of NDD_u^2 to define the node u's NDD signature. A distance q of 2 is sufficiently revealing for social networks which are known for having a small average path length: larger values of q will end up recording a large part of the graph which leads to high redundancy in training data. We use a bin size of 50, which was shown empirically [9] to capture the high degree variations of large social graphs. For each q, we use 21 bins, which would correspond to a larger node degree of 1050. All larger values are binned in to the last bin. This binning strategy is designed to capture the aggregate structure of ego networks [43].

Note that the nodes in $G_{san} \cap G_{aux}$, common to both graphs, can be recognized as being the same node (identical) in the two graphs based on their node identifier. Non-identical nodes are unique to each G_{san} and G_{aux} and do not exist in the overlap. We use a learning algorithm based on an ensemble of random decision trees (i.e., Random Forest) to perform the classification task of quantifying graph vulnerability [63]. The classification task outputs 1 for identical node pairs and 0 for non-identical node pairs. This is the ground truth against which we measure the accuracy of the learning algorithms. We generate examples for the training phase of the de-anonymization attack by randomly picking node pairs from the sanitized (G_{san}) and the auxiliary (G_{aux}) graphs,

respectively. Each training example represents a pair of nodes (each node being represented by its NDD) and whether the nodes are identical or not.

In most cases, we have an unbalanced dataset with the degree of imbalance depending on the overlap parameter α, where the majority is non-identical node pairs. We use the reservoir sampling technique [64] to take ℓ=1000 balance sub-samples from the population S, and the SMOTE algorithm [65] as an over-sampling technique for each sub-sample. Each sample is trained by a forest of $\jmath$=100 random decision trees. Each decision tree performs a binary classification to measure the quality of the classifier on the task of differentiating two nodes as identical or not. We use both *bagging* [66] and *randomized node optimization* [67] techniques to select a random subset of training examples with a random subset of features for each learner to train and test respectively. Having many decision tree learners enable us to mount multiple attacks in the same graph space. Therefore, we devise $\ell \times \jmath$=100,000 attack scenarios per one input graph.

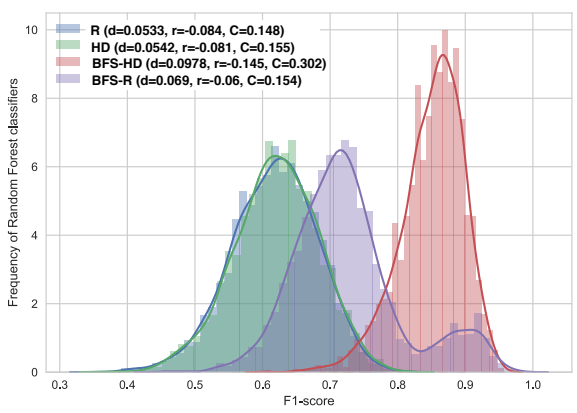
We measure the accuracy of the classifier in determining whether a randomly chosen pair of nodes (with one node in G_{san} and another in G_{aux}) are identical. We use F1-score to evaluate the quality of the classifier. F1-score is the harmonic mean between precision and recall, typical metrics for prediction output of machine learning algorithms. For each data sample, we perform 5×2 cross-validation to evaluate the classifier and record the mean F1-score.

Intuitively, the strength of an attacker is not solely defined by the size of the subgraph to which the attacker has access, but also by the "quality" of the subgraph [53]: for example, a disjoint set of low degree nodes (which would be the case of a randomly chosen set of nodes from a power-law graph) carries less structural information than a connected subgraph of the same number of nodes. Figure 2.2 presents the performance of the node re-identification attack under different methods of

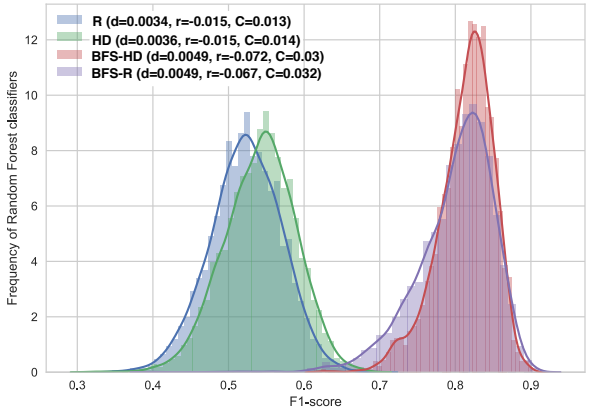
building the overlap subgraph. We present these results here before we introduce the datasets for two reasons. First, the results are consistent across all datasets tested, and they support the intuition presented above. Therefore, the characteristics of the datasets are irrelevant for understanding these particular experimental results. Second, we only present these results to justify our choice for building the overlap subgraph in the rest of the experiments. To maintain the reading flow, we present all design details in this section.

Figure 2.2 confirms multiple intuitions. First, the attack is consistently and significantly stronger when the nodes in the overlap are connected (scenarios marked with BFS in the plot). Second, the attack is stronger when the density of the overlap is higher.

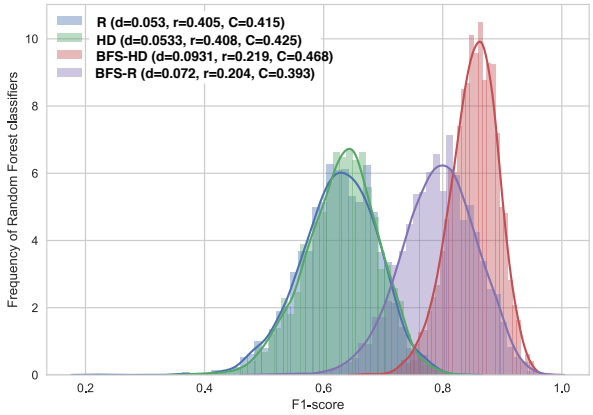
In the rest of our study, we use BFS-HD to generate the overlap. In addition to being a stronger attack because of starting from a richer knowledge set, our attack mechanism based on BFS-HD turns out to be representative for *percolation-based* network alignment methods [51] proposed in other contexts, such as protein-protein interaction networks. Also, many de-anonymization attacks [47, 68, 8] employ similar techniques based on the percolation theory. Our machine-learning based attack is thus a generalization of existing de-anonymization attacks that have the same core ingredients: start from a set of already identified nodes and successively identify their neighbors. The reason behind the success of ML-based de-anonymization techniques is that they learn automatically invariants useful for node re-identification. Thus, the same ML-based de-anonymization attack can be used successfully against different anonymization techniques [9].



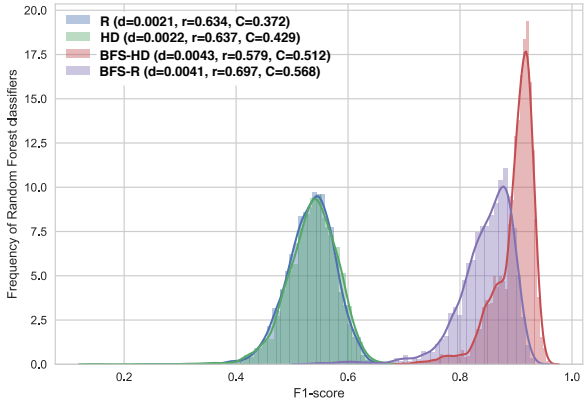
(a) fb107: $1K$



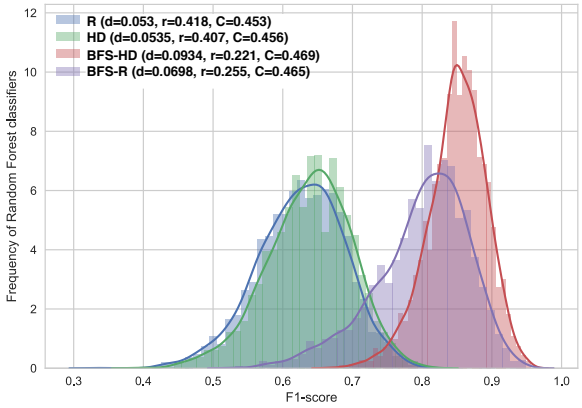
(b) caGrQc: $1K$



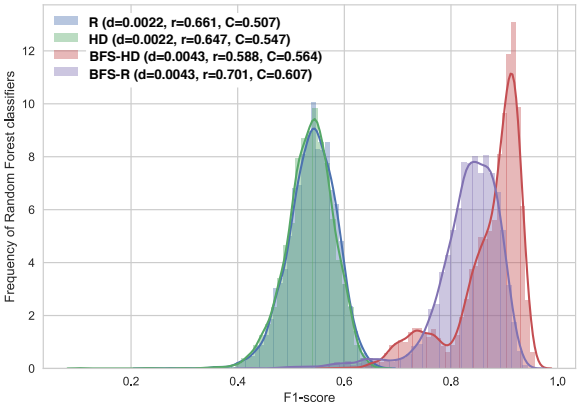
(c) fb107: $2K$



(d) caGrQc: $2K$



(e) fb107: $2.5K$



(f) caGrQc: $2.5K$

Figure 2.2: A comparison of attack strength based on different overlap choices. The overlap choices include Random (R), High Degree (HD) and BFS-trees (rooted in the highest degree node BFS-HD and, respectively, a random node, BFS-R). Accuracy of predicting identical pairs is presented over different dK spaces. Graph properties of density ($\bar{d}$), assortativity (r), and transitivity (C) are averaged over 8 subgraphs per dK-space that are associated with the given overlap.

2.2.3 Causality Analyzer

The objective of the Causality Analyzer component is to reveal the effect of topological metrics on graph vulnerability.

In our implementation, we chose to study the causality and associativity relationships [69]. In both cases, we start from the same set of metrics, vulnerability (as measured by the F1-score) and a set of graph measurements obtained with classical social network analysis techniques, and apply different tools to isolate the strength of the causality and the strength of associativity relationships. As before, these tools can be replaced with different ones than we employed here.

2.2.3.1 Causality via Explanatory Modeling

We use explanatory modeling techniques [70] to measure the significance of the causal relationship between graph metrics and graph vulnerability.

We estimate the graph vulnerability function f through several regression tests, both linear and entropy-based. Each model tests the individual effect of graph metrics on explaining the graph vulnerability score. The target variable is the vulnerability score (i.e., F1-score), and independent variables are the associated structural properties including macro-level graph metrics such as *density, assortativity, transitivity, average path length, and the proportion of degree-1 nodes* in the given network. We select these properties as an example for studying the importance of community structure on the success of node re-identification attacks.

We use F-test [70] and Mutual Information (MI) to measure the causality in the relationship [71]. F-test captures the significance of any independent variables on the correlation with the target variable using multiple linear regression models. MI is defined as a nonlinear function of

the join probability measure between target variable and independent variables, which captures any kind of dependency in the variable space. Both F-test and MI are in the range of 0 to 1, and the higher values represent more significant dependencies.

In order to infer the causal relationship in the target and independent variable space, we use the Pearlian framework [72] which produces a Directed Acyclic Graph (DAG) describing the causality. Specifically, we use the IC* (Inductive Causation) algorithm [72] in the Pearlian framework to establish this causality. The core idea is to find whether variable X has a (direct) causal influence on variable Y. This algorithm outputs a directed acyclic graph where each variable represents a node, and the edge represents a statistical dependence between variables for causation. The IC* algorithm recursively constructs this graph after performing probabilistic conditional independence tests for all pairs and triplets of variables. Specially, the algorithm assumes the existence of hidden confounding variables when deriving latent causal structure. This is important as we do not cover the entire graph metric space in our analysis. We used an open-source Python implementation [73] of the Pearlian framework to perform the causality inference.

2.2.3.2 Associativity via Predictive Modeling

As an attempt to uncover potential association between graph metrics and vulnerability, we quantify the level of predictability of the vulnerability score using structural properties. We construct the graph vulnerability function f from the examples of derived vulnerability scores associated with structural properties in the given graph. Note that we use the same set of structural properties used in explanatory modeling. Our models assess the ability to generate predictions of vulnerability score for a set of unseen graphs, when given only the structural metrics as graph descriptions. On validating predictive models, we provide measurements related to the *residuals*

and *generalization* of each model. We use cross-validation (i.e., holdout sets) techniques to avoid over-optimistic performance of the prediction results.

We report three metrics to measure the regression performance. First, the *Root Mean Squared Error (RMSE)*, which corresponds to the expected value of squared loss in the same units as the target variable. Second, *Explained Variance (EVAR)*, which measures the significance of the variance of the error with respect to the variance of the target variable. Finally, the R^2 *score* (R^2S), which measures the likelihood of predicting future examples correctly. RMSE ranges from 0 to ∞, where lower values in the range of F1-score ($0 \leq$ F1-score ≤ 1) depict more accurate predictions. $EVAR$ and R^2S range from $-\infty$ to 1, the higher the values, the more accurate the models.

2.3 Datasets

Our objective is to evaluate the framework we proposed for quantifying what structural properties make graphs more vulnerable to de-anonymization attacks. To this end, we select a number of real network datasets (presented in Section 2.3.1) and generate families of synthetic graphs using three different approaches that control particular graph metrics (as presented in Section 2.3.2). These synthetic graphs serve to capture both independent and inter-dependent structural forces in the network.

2.3.1 Real World Networks

We chose four publicly available datasets that represent real social networks of various types. *fb107* [74] represents social circles of an ego in Facebook. *caGrQc* [75] is a co-authorship network between the authors of papers in general relativity and quantum cosmology. *soc-anybeat* [76] is an interaction network available in the Anybeat online community, which is a public gathering place

across the world. Finally, *soc-gplus* [77] is a follower network from Google+. Table 2.1 summarizes the properties of these datasets.

2.3.2 Synthetic Networks

In order to be able to control graph characteristics, we also generated families of synthetic graphs with the subsets of the characteristics of the real datasets. We used three graph generation techniques that individually cover different spaces of graph metrics.

dK-Random graphs model topological constraints systematically with respect to the node degree. They are known to be less random and more structured the higher the d (presented in Section 2.3.2.1). While degree distributions have been shown to capture very important graph properties [78], they typically fail to reproduce some others, such as the clustering coefficient. In order to analyze graphs with controlled clustering coefficient (that is, similar to those of the real networks studied), we employ the second graph generation technique: The Exponential Random Graph Model (ERGM) is a mature modeling framework that maximizes the likelihood of generating a random graph with given properties (presented in Section 2.3.2.2).

While widely used especially in Sociology, in our experience ERGMs fail to generate graphs with the desired range of degree assortativity coefficient. In order to vary this structural characteristic and cover the corresponding graph space, we used another technique specifically designed to generate graphs with a good range of local and global assortativity coefficients, a model that we name the Leader-Follower (LF) model and present in Section 2.3.2.3.

2.3.2.1 dK-Random

The dK-series represents a set of descriptive statistic metrics that capture the original graph structure at multiple levels of detail [79, 41]. Specifically, the dK-series summarizes the structure of a graph from the degree distribution of a subgraph pattern of size d. Thus, $0K$-graphs are random graphs with a given average node degree, $1K$-graphs are random graphs with a given degree distribution, $2K$-graphs are random graphs with a given joint degree distribution, $3K$-graphs are random graphs with a given interconnectivity of triplets of nodes, and so on. Intermediate steps in the series can be defined, such as the $2.5K$ graph, which is a relaxed version of $3K$-graphs that reproduces both joint degree distribution and degree-dependent clustering coefficient [80]. We used RandNetGen [78] to generate $0K$, $1K$, and $2K$ graphs. (In this work we have not used 2.5 graph generators, as controlling the clustering coefficient independently from the joint degree distribution better fits our objectives). No graph generative algorithms are known for steps higher in the series [78].

2.3.2.2 ERGM

Exponential-family random graph models (ERGMs) or p-star models [81, 82] are used in social network analysis for stipulating, within a set structural parameters, distribution probabilities for networks. Its primary use is to describe structural and local forces that shape the general topology of a network. This is achieved by using a selected set of parameters that encompass different structural forces (e.g., homophily, degree correlation/assortativity, clustering, and average path length). Once the model has converged, we can obtain maximum-likelihood estimates, model comparison and goodness-of-fit tests, and generate simulated networks tied to the relationship between the original network and the probability distribution provided by the ERGM.

Our interest in ERGMs is based on simulating graphs that retain set structural information from the original graph to generate a diverse set of graph structures. We used R [83] and the *statnet* suite [84], which contains several packages for network analysis, to produce ERGMs and simulate graphs from our real-world network datasets. In this case, we focused on three structural aspects of the graphs: clustering coefficient, average path length, and degree correlation/assortativity. For the ERGM based on clustering coefficient (*ERGM-cc*), we used the *edges* and *triangle* parameters in the statnet package. The *edges* parameter measures the probability of linkage or no linkage between nodes, and the *triangle* term looks at the number of triangles or triad formations in the original graph. For the average path length model (*ERGM-apl*), *edges* and *twopath* terms were used. The *twopath* term measures the number of 2-paths in the original network and produces a probability distribution of their formation for the converged ERGM. Lastly, for the assortativity measure (*ERGM-dc*), the terms *edges* and *degcor* were used to produce the models. The *degcor* term considers the degree correlation of all pairs of tied nodes (for more on ERGMs see [85, 86]). These terms proved to be our best choices for preserving, to a certain extent, the desired structural information. Although the creation of ERGMs is a trial and error process, the selected terms were successful in producing models for each of the original networks.

2.3.2.3 Leader-Follower

We use Leader-Follower (LF) model [87] to generate networks with controlled degree-based assortativity coefficients. This model controls two node populations in which one group (i.e., followers) selects edges randomly to connect such that the preferential attachment behavior emerges spontaneously, while other group (i.e., leaders) adopts an anti-preferential behavior which creates ties to lower-degree nodes. The generation algorithm requires three parameters: p is the fraction

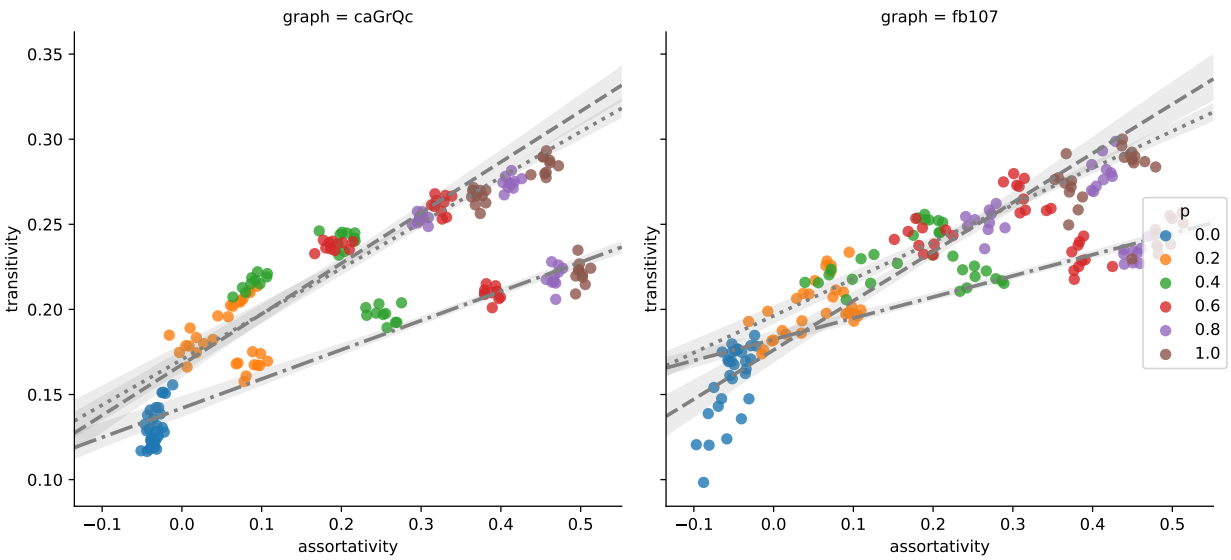


Figure 2.3: Transitivity (C) and assortativity (r) on LF graphs. Multiple regression models are presented as a function of m, where *dashed*, *dotted* and *dash-dotted* lines represent models for $m = 2, 5$, and 10, in this order.

of leader nodes, m is the maximum number of connections possible for a node to initiate, and l defines the extent of neighborhood information available for a node to decide initial connections. For simplicity, we set $l = 1$, such that a new node decides its choices to connect from an immediate neighborhood around an anchor node.

When there are no leader nodes ($p = 0$), the generated networks exhibit strong preferential attachment behavior, leading to a negative degree-assortativity value. When $p = 1$, the resulting graphs have positive degree assortativity. Experimentally, p was confirmed to be proportional to degree assortativity, as shown in Figure 2.3 for two of the real datasets we analyzed. Note also the linear relationship between transitivity and degree assortativity in this plot.

Table 2.1: Graph properties of the real and synthetic network datasets. All graphs are undirected. Density ($\bar{d}$) is the fraction of all possible edges. Degree-assortativity (r) measures the similarity of relations depending on the associated node degree. Transitivity (C) is the fraction of triangles of all possible triangles in the network. Average path length (κ) depicts the average shortest path length between any pairs of nodes and degree-1 represents the percentage of nodes in the network with degree exactly 1. Average values are presented over 100 synthetic graphs per space.

Network	space	$\|N\|$	$\|E\|$	$\bar{d}$	r	C	κ	degree-1 (%)
fb107	original	1034	26749	0.0500	0.4316	0.5045	2.9517	1.45
	0K	1034	26749	0.0501	-0.0029	0.0501	2.0210	0.0
	1K	1034	26749	0.0501	-0.0961	0.1466	2.1965	1.45
	2K	1034	26749	0.0501	0.4316	0.3161	2.4020	1.45
	ERGM-apl	1034	26749	0.0501	0.0017	0.0504	2.0193	0.0
	ERGM-cc	1034	26749	0.0501	0.4293	0.5038	2.8796	0.57
	ERGM-dc	1034	26749	0.0501	0.3747	0.1627	2.1197	0.0
	LF (m=2)	1034	2066	0.0039	0.1425	0.2173	10.2155	0.0
	LF (m=5)	1034	5165	0.0097	0.2308	0.2463	5.5336	0.0
	LF (m=10)	1034	10330	0.0193	0.2733	0.2164	3.6806	0.0
caGrQc	original	5242	14496	0.0011	0.6592	0.6298	3.8047	22.83
	0K	5242	14496	0.0011	-0.0011	0.0010	5.2155	2.22
	1K	5241	14484	0.0011	-0.0355	0.0077	4.0002	22.83
	2K	5241	14484	0.0011	0.6593	0.2710	1.0410	22.83
	ERGM-apl	5241	14484	0.0011	0.0390	0.0064	5.4390	0.02
	ERGM-cc	4507	14484	0.0014	0.6804	0.6278	5.6361	10.43
	ERGM-dc	5237	14484	0.0011	0.4547	0.0790	5.5294	0.98
	LF (m=2)	5242	10482	0.0008	0.1536	0.2132	13.0612	0.0
	LF (m=5)	5242	26205	0.0019	0.24	0.2348	7.1527	0.0
	LF (m=10)	5242	52410	0.0038	0.2771	0.1895	4.7513	0.0
soc-anybeat	original	12645	49132	0.0006	-0.1234	0.0217	3.1715	49.51
	0K	12645	49132	0.0006	-0.0001	0.0006	4.8365	0.33
	1K	12645	49132	0.0006	-0.1232	0.0149	2.8779	49.50
	2K	12645	49132	0.0006	-0.1234	0.0176	2.4943	49.50
	ERGM-apl	12635	49132	0.0006	-0.0572	0.0018	3.2206	0.61
	ERGM-cc	12582	49132	0.0006	0.2285	0.1877	4.9853	2.57
	ERGM-dc	12459	49132	0.0006	-0.0831	0.0158	3.3204	8.93
soc-gplus	original	23628	39194	0.0001	-0.3885	0.0037	2.2082	69.16
	0K	23628	39194	0.0001	0.0009	0.0001	7.7045	12.46
	1K	23628	39194	0.0001	-0.3514	0.0137	3.1760	69.16
	2K	23628	39194	0.0001	-0.3885	0.0018	3.8620	69.16
	ERGM-apl	22544	39194	0.0001	-0.0729	0.0004	4.5236	15.32
	ERGM-cc	17784	39194	0.0002	-0.0651	0.0337	5.8122	39.76
	ERGM-dc	22042	39194	0.0001	-0.2407	0.0024	4.0795	30.52

2.4 Empirical Results

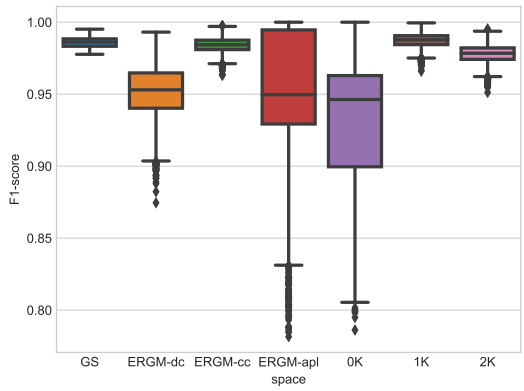
Our objectives for empirical evaluations are twofold. On one hand, we aim at evaluating the utility of the framework we proposed. On the other hand, we use the framework to answer the question: *What structural properties makes some graph datasets more vulnerable to attacks than others?*

We start by evaluating the vulnerability of the real and synthetic graphs in our collection. We quantify the vulnerability of a graph as a function of the rate of successful node re-identification, and present a comparison of vulnerability scores across different families of graphs (as presented in Section 2.4.1). Furthermore, we perform a rigorous analysis on the relationship between graph vulnerability and different structural forces to identify the factors that contribute towards a successful de-anonymization attack. We present both information-theoretic (as presented in Section 2.4.2) and performance (as presented in Section 2.4.3) measurements to evaluate this relationship.

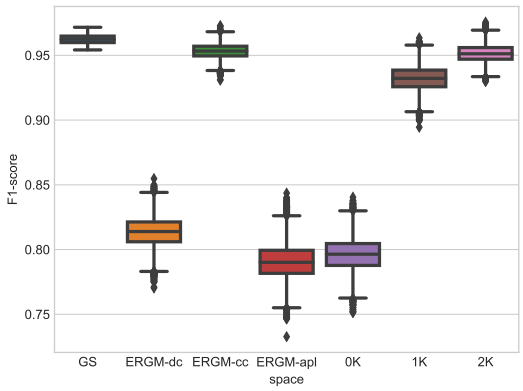
2.4.1 Graph Vulnerability Analysis

We report the F1-score as the accuracy of predicting the structural equivalence of a pair of nodes, which we refer to as graph vulnerability score. Figure 2.4 presents a comparison of vulnerability scores for different synthetic graph spaces. We observe three phenomena.

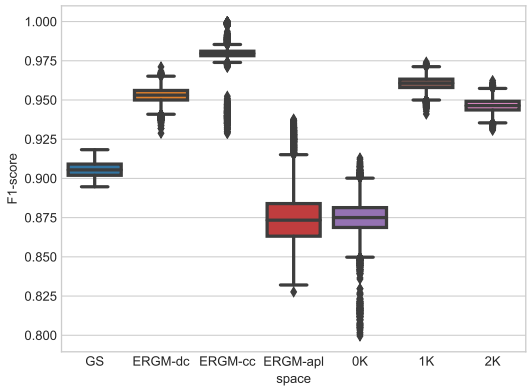
The first observation is related to the comparison of vulnerability scores in ERGM spaces. The mean vulnerability score increases in the order of *ERGM-apl*, *ERGM-dc* and *ERGM-cc*, while *ERGM-apl* shows the widest range. What this seems to suggest is that preserving assortativity and transitivity as utility metrics in an anonymization technique can potentially damage the anonymity of the nodes in the graph. To the best of our knowledge, we are the first to observe this phenomenon.



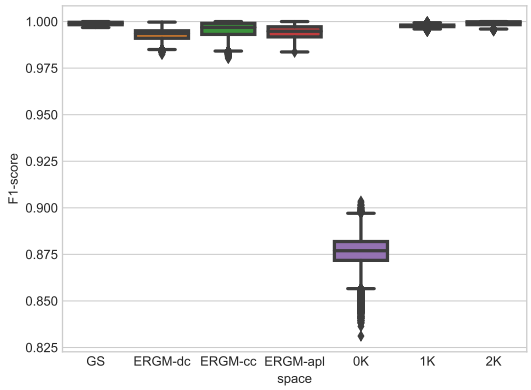
(a) fb107



(b) caGrQc



(c) soc-anybeat



(d) soc-gplus

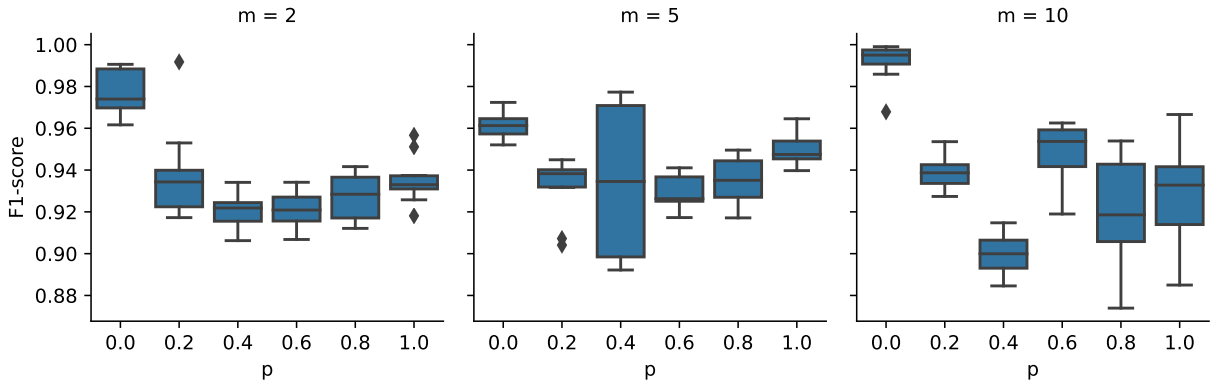
Figure 2.4: A comparison of the F1-score over different graph input spaces. Each score represents prediction results of 5×2 cross-validation samples, which is averaged over 100 synthetic graphs per space.

To better understand the effect of degree assortativity, we focus on LF-generated graphs, where assortativity is varied. Figure 2.5 presents a comparison of graph vulnerability scores as a function of the LF graph generator parameters, p and m, as presented in Section 2.3.2.3. The vulnerability score reaches a local maximum for small p and drop to local minima when p is in the range of $0.4-0.6$. Since p is proportional to assortativity, which in turn is proportional to transitivity for the given LF graphs (Figure 2.3), it is highly likely that assortativity and transitivity are factors

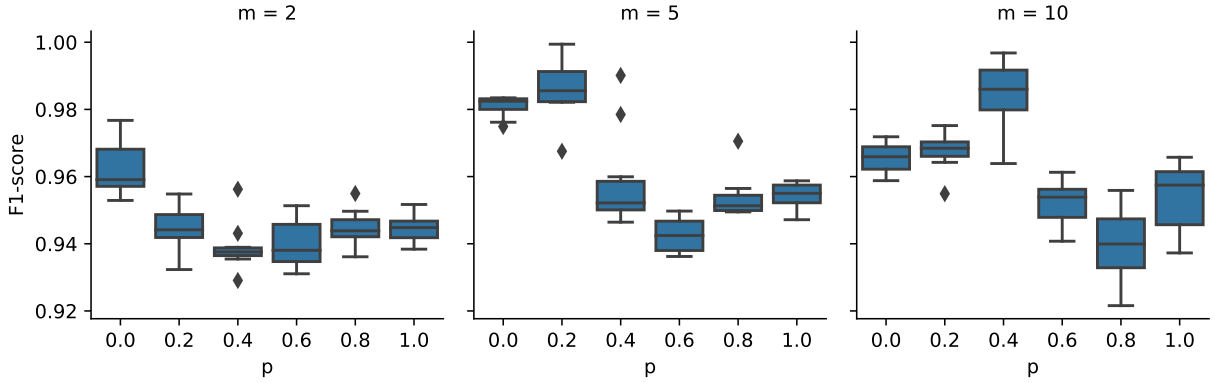
of graph vulnerability. We perform a rigorous statistical analysis in the next section to understand how assortativity and transitivity affect the graph's vulnerability to de-anonymization attacks.

Second, as Figures 2.4a–2.4d show, some of ERGM-generated graphs are more vulnerable than 1K or even 2K graphs despite the fact that they do not replicate the original graph distribution. From previous results, the intuition was that privacy increases with the perturbation of the degree distribution [41]. Our results show that a different graph metric—in this case transitivity—is even more revealing than the degree distribution. Specifically, in the *soc-anybeat* network, the vulnerability of the ERGM-cc generated graphs is higher on average than the average vulnerability of the 1K and 2K graphs (Figure 2.4c). This is happening despite the fact that the ERGM-cc graphs have a very different degree distribution, as seen in the last column of Table 3.1: while the original graph (and thus the degree-preserving 1K and 2K graphs) had 49.5% of nodes with degree 1, the ERGM-cc has only 2.57% such nodes. This result shows that there are structural properties that make a graph more vulnerable to re-identification attacks than the degree distribution does. While previous work [41] showed that disturbing the degree distribution is necessary for anonymity, we show that it not sufficient: other graph metrics must also be perturbed to achieve anonymity.

Third, some known phenomena are confirmed by our experiments. The original graph (denoted as GS) is more vulnerable in all cases, except for the *soc-anybeat* network (Figure 2.4c). We discuss the reason behind this divergence later. At the other end of the spectrum, $0K$ (or Erdös-Rènyi) graphs are (as expected) the least vulnerable, but also the least representative of real datasets. In addition, the vulnerability scores of $1K$ and $2K$ graphs are the closest to the original. This confirms already known results that show that dK graphs lack real expectations of privacy, since higher dK graphs leak significant graph structural information [88, 41].



(a) fb107



(b) caGrQc

Figure 2.5: A comparison of the F1-score over LF graphs. We generated a number of LF graphs by varying parameters p and m. p is the probability that the network exhibits a force towards anti-preferential attachment, which is positively correlated with degree-assortativity(r), while m is proportional to density($\bar{d}$) of the network. Each score represents prediction results of 5×2 cross-validation samples, which is averaged over 10 synthetic graphs under the parameters of p and m.

2.4.2 Causality Analysis Based on Explanatory Modeling

While Figures 2.4 and 2.5 show high variation in vulnerability with different topological constraints, it is impossible to visually conclude what makes a graph more vulnerable. We study the dependencies between the graph vulnerability score and a set of macro-level structural graph properties to identify such patterns. One such pattern, in fact a causal explanation for graph vulnerability, is presented in Figure 2.7 using explanatory modeling techniques. Two metrics of

importance are used: F-test and mutual information (MI), as described in Section 2.2.3.1. In the dK and ERGM spaces (as shown in Figures 2.7a and 2.7b), assortativity shows a relatively low F-test value, suggesting a weak linear dependency with the vulnerability score. Meanwhile, it shows significantly higher MI value, which suggests a better reduction of uncertainty on explaining the vulnerability score. Transitivity also appears more non-linearly dependent in the dK space, since MI is relatively higher. The average shortest path length has mixed results in F-test, but MI reaches maximum for both spaces. The proportion of degree-1 nodes is shown to be a strong candidate of dependency with graph vulnerability. It shows higher values for both F-test and MI. This somewhat explains the position of the original *soc-anybeat* network (GS) with respect to the vulnerability score in Figure 2.4c. Comparing with generated ERGM graphs, *soc-anybeat* original graph has 49.5% degree-1 nodes who are structurally indistinguishable. However, degree-1 nodes also reveal less information about their neighbors' positions in the network.

Figure 2.7c presents the dependency analysis of LF graphs. Similar to dK and ERGM graph spaces, assortativity and transitivity show relatively higher MI values, and average shortest path length reaches the maximum MI. It appears that transitivity is a linear function of vulnerability (F-test=1). In fact, when we control for assortativity, transitivity is found to be positively correlated with assortativity in LF graphs (Figure 2.3). Since transitivity is linear with vulnerability and assortativity is linear with transitivity, we would expect assortativity to linearly cause vulnerability. However, this is not the case, as shown by F-test=0.53 in the second plot of Figure 2.7c.

Figure 2.6 shows the pictorial view of the Pearlian Directed Acyclic Graph (as presented in Section 2.2.3.1) which describes the causal pathways from one graph metric to another, or to the graph vulnerability score derived from our experimental data. We do not specify any prior

assumption about the causality in the Pearlian framework, but let the IC* algorithm to decide the optimal causal pathways based on probabilistic conditional independence tests. In Pearlian Directed Acyclic Graph, bidirected edges represent indirect casual relationships due to unobserved variables.

We observe two phenomena: First, transitivity, density, average path length and the fraction of degree-1 nodes have a direct statistical dependency with the graph vulnerability score. However, such dependencies are not identified as genuine causal relationships by the Pearlian framework. This set of dependencies could be due to a set of other (unobserved) *confounding* graph metrics. In our setting, a *confounding* metric presents an alternative explanation for the observed statistical dependency between a graph vulnerability and the associated graph metrics. While average path length has an immediate *confounding* effect on the causal pathway between transitivity and graph vulnerability, the fraction of degree-1 nodes and assortativity have shown a combined *confounding* effect for the same causal pathway.

Second, assortativity does not have a direct statistical dependency with the graph vulnerability score, but has *confounded* other graph metrics (i.e., transitivity, density and the fraction of degree-1 nodes) to cause an effect on graph vulnerability. In general, this *confounding* effect from assortativity is well captured by transitivity, and it transforms to cause an effect on graph vulnerability.

In conclusion, vulnerability can be explained as a linear function of the fraction of degree-1 nodes, and a non-linear function of other graph metrics. Non-linearity of the relationship between transitivity, assortativity, and graph vulnerability score is being significantly highlighted by the explanatory modeling techniques we used. In the next section we further analyze this relationship over the predictive capability of graph metrics.

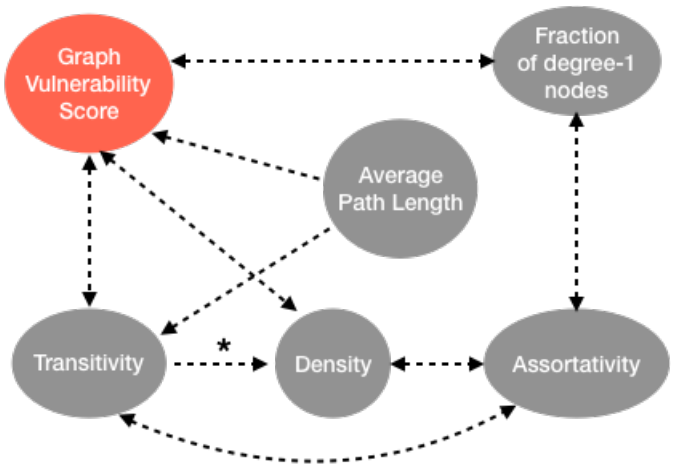
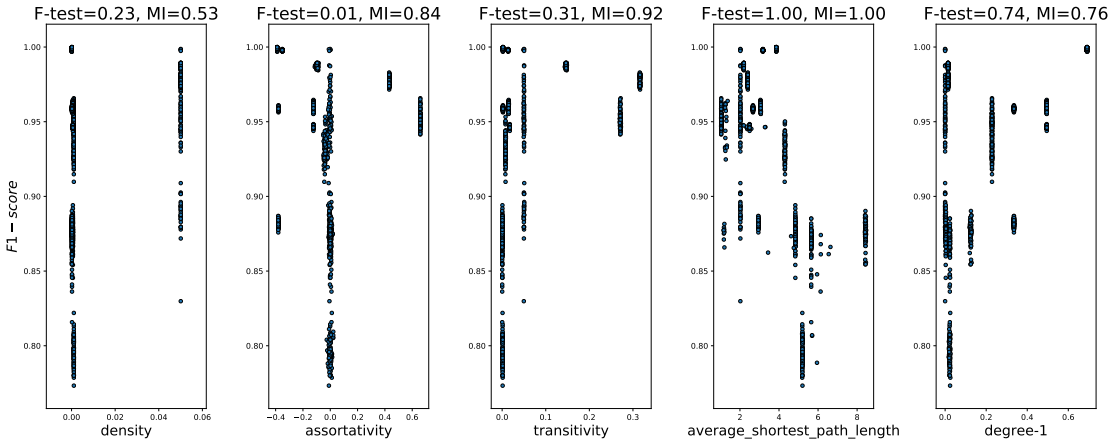


Figure 2.6: Pearlian directed acyclic graph. We use the Pearlian framework [72] for the causal inference over the graph metrics and graph vulnerability score. The edge direction represents the cause-effect relationship, where the arrow head points to the effect. The "*" notation on the edge indicates the belief of the causal inference algorithm about the genuine causal relationships.

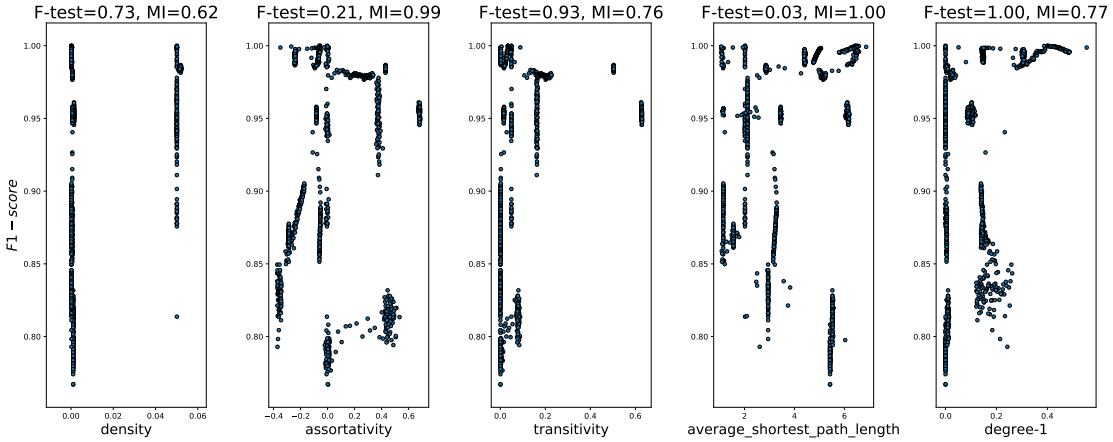
2.4.3 Performance Analysis Based on Predictive Modeling

So far, we analyzed the relationship between graph vulnerability score and associated graph metrics without making any assumptions of a prediction model. Though such analysis reveals important insights, the observations could not be generalized for any collection of networks. Our framework supports another set of measurements based on predictive modeling. We fit the examples into multiple regression models, and report the accuracy on predicting the vulnerability score for an unseen set of graphs. The target variable is the F1-score, and the feature space includes the same set of structural properties that we studied earlier. Section 2.4.3.1 and 2.4.3.2 report the accuracy for linear and polynomial regression models, respectively.

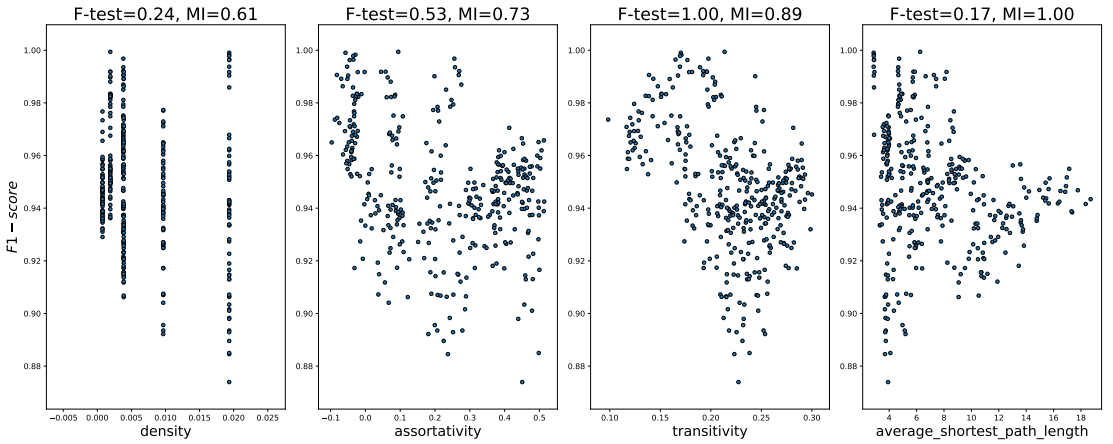
We prepare holdout sets of examples in two ways. First, we split data based on graph spaces, and create three folds of data: dK, ERGM and LF. Then we perform *3-fold cross validation*, and report the performance of predictions. As an example, we train on dK space examples and test on ERGM space examples.



(a) dK



(b) ERGM



(c) LF

Figure 2.7: A comparison between F-test and mutual information measures. We calculate F-test and mutual information between the vulnerability of a graph and associated structural properties across different graph input spaces.

Table 2.2: A comparison of the accuracy of predicting F1-score. We use linear regression model to predict F1-score on different cross-validation graph spaces using structural properties including density, assortativity, transitivity, average shortest path length and the percentage of degree-1 nodes.

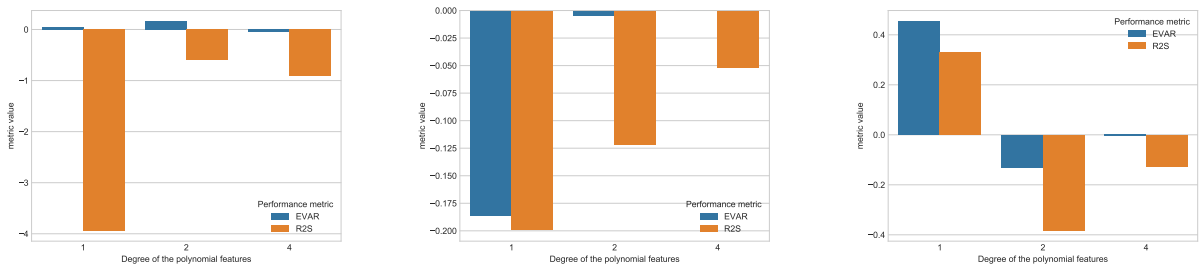
Training	Testing	RMSE	EVAR	R^2S
Synthetic	Original	6.3156	-0.0020	-2.3627
Synthetic	Synthetic	0.0821	0.0578	-3.9392
dK	dK	0.0533	0.2162	-4.5240
dK	ERGM	0.0753	-0.1867	-0.1994
dK	LF	0.0392	-0.2170	-0.4052
ERGM	ERGM	0.1372	-1.4304	-48.9614
ERGM	dK	0.0646	0.4552	0.3288
ERGM	LF	0.0297	-2.7625	-2.9118

Second, we split data based on the families of original graphs for the predictions in the same graph space, such that we create five folds of data, one for each family of original datasets. Then we perform *5-fold cross validation*, and report metrics on average. For example, one instance of cross-validation includes training on examples from dK graphs generated from four networks, and tests on examples from dK graphs generated from the fifth network. We repeat the same process for ERGM graphs as well as all synthetic graphs.

2.4.3.1 Linear Regression Model

Table 2.2 presents the residuals and the coefficients of determination for the predictions across different cross-validation sets of data. We make three observations.

First, when traning and testing take place within the same space (i.e., train on the synthetic spaces defined by four original datasets and test on the synthetic graph space of the fifth dataset), the dK space enables better accuracy than the ERGM space. Moreover, the dK space enables better accuracy than when training is done on all synthetic spaces. What this means is that the dK space is more vulnerable to re-identification attacks than any of the other spaces. For example, the RMSE



(a) Train: Synthetic and Test: Synthetic(b) Train: dK and Test: ERGM (c) Train: ERGM and Test: dK

Figure 2.8: The performance metrics over the degree of polynomial features. We generate polynomial and interaction features from original feature space describing graph structural properties, such that the new feature space includes all polynomial combinations of features under the specified polynomial degree. Performance metrics are based on the associated polynomial regression model.

value for cross-validation in the dK space (0.05) is lower than in the ERGM space (0.13). Also the variation of the predicted vulnerability score is better explained from the features in dK than ERGM space regression model, where $EVAR_{dK-dK} = 0.21 > 0$ and $R^2S_{dK-dK} > R^2S_{ERGM-ERGM}$. Note that the set of subscripts represents the training and testing set in the consecutive order.

Second, training on dK spaces and testing on ERGM spaces performs poorly compared with the case where the training is done on the ERGM space and the testing on the dK space, where $EVAR_{ERGM-dK} > EVAR_{dK-ERGM}$ and $R^2S_{ERGM-dK} > R^2S_{dK-ERGM}$ (see Table 2.2). This suggests that the ERGM space is a richer training dataset than the dK space, which means that the synthetic graphs in this space have more variation in the graph metric values and vulnerability. In other words, the dK space constrains more drastically the values of the graph features considered, and thus limits the learning. This behavior is the result of the dK space definition, but may also be the outcome of the particular dK random graph generator we used.

And third, dK is a better training set for testing on LF than the ERGM space is. This is likely explained by the fact that the LF and the dK spaces are closer to each other than they

are to the ERGM space. Specifically, the degree assortativity (which controls the LF space) is an aggregate measure of the joint degree distribution that defined the dK space. Moreover, the LF space is generated only by two datasets, both with high assortativity and clustering. The good predictability from training on the dK space confirms again the effect that assortativity and clustering have on graph vulnerability.

2.4.3.2 Polynomial Regression Model

In our discussion so far, we outlined the linearity of structural properties with respect to the given vulnerability. In general, our observations suggest there is a non-linear relationship with the target and independent variables. We try to account for such a relationship through a polynomial regression model. First, we transform the features (i.e., structural properties) to a new polynomial feature space. This new space includes all polynomial combinations of raw structural property values, and all interaction terms. Figure 2.8 presents the predictive power using two metrics of interest: EVAR and R^2S (explained in Section 2.4.3), specifically to understand the variance of graph vulnerability scores through the set of structural properties. We compare several regression models in three polynomial feature spaces, under linear, quadratic, and quartic polynomial degrees. Linear space is similar to the results we presented earlier in Section 2.4.3.1.

We make a number of observations. In the *Synthetic* training model, R^2S increases significantly in the quadratic polynomial space (Figure 2.8a). EVAR also reaches the local maximum in this polynomial degree space. (Note that our synthetic space includes both dK and ERGM generated graphs, and average values are calculated over different cross-validation tests.) This proves the existence of a combination effect of structural forces on explaining graph vulnerability.

For a finer view of the graph vulnerability based on different synthetic spaces, we present two local analyses: one related to the dK training model (Figure 2.8b) and the other related to the ERGM training model (Figure 2.8c). In both training models, the variation of the target variable (vulnerability) is better captured when increasing the degree of polynomial features (since both EVAR and R^2S increase). However, we observe a special case in ERGM training model (as presented in Figure 2.8c). The predictive power of the linear model is weakened after the addition of interactive terms in quadratic space, which does not happen in the dK training model. In a sense, we relax the utility conserved in ERGMs by transforming to a different feature space, thus having relatively worst predictive model.

2.5 Summary and Discussions

This chapter poses and answers a new research question: *What graph properties make network datasets more vulnerable to node re-identification attacks?* Unlike previous related research, we question the intrinsic vulnerability of an original graph dataset rather than any particular anonymized version of the dataset. An answer to this question can be used both to asses the risk of publishing an original dataset and also to guide the data practitioner in selecting anonymization techniques that provide the appropriate tradeoff between utility and privacy.

We introduce and experiment with a framework that identifies the relationships between graph vulnerability and graph properties. Our code is available for download at [42]. The components of this framework include i) a quantification of graph vulnerability as measured by the success of a re-identification attack; ii) a quantification of the relationship between graph vulnerability and a set of graph metrics. Moreover, we instantiated this framework with a strong attack model and

a rigorous set of tools for causality analysis. Using thousands of synthetic graphs of controlled properties we discovered a number of phenomena.

First, under the attack model considered, there is a strong statistical dependency between the vulnerability score and transitivity and assortativity. That is to say, successful anonymization techniques should not attempt to preserve the assortativity and transitivity of the original graph. In other words, one could design an anonymization technique to explicitly perturb assortativity and transitivity for increasing graph privacy. This observation opens a new door for designing anonymization algorithms that has a chance against strong de-anonymization attacks.

Second, there is no linear relationship between the vulnerability score and the graph metrics other than the fraction of degree-1 nodes in the network. One reason is that the most relevant graph metrics in network analysis are interdependent [78]. Using a larger number of graph metrics in the Pearlian causality model should help identify a more complex causal relationship between graph vulnerability and properties.

Third, our comparison across graphs generated by different graph model generators lead to an important conclusion. In an early work, Hay et al. [11] observe a graph's density as a determinant to describe the asymptotic limit of graph vulnerability. It was also well understood that preserving the degree distribution or the degree correlation increases graph vulnerability [41] and thus disturbing them is a necessary condition for graph anonymization. However, our study shows that this condition is not sufficient: in some cases, other network properties independent of the degree distribution put node privacy at risk. This is a disturbing result for the current understanding of graph privacy, as it means that protecting graph privacy is much harder than previously considered [10, 11].

One concern is whether some of our observations depend on the tools we used to implement the components of our framework. Specifically, we mounted a strong de-anonymization attack that led to high vulnerability scores. One could argue that a different attack model or a different feature representation for nodes (weaker than the NDD representation we used) could lead to different vulnerability scores that might indicate a different relation between graph vulnerability and graph metrics. We believe this is a valid concern and it highlights the usefulness of the framework we propose. For example, one could use our framework to derive the causal relationship between the parameters of an attack model and the rest of variable space including the graph vulnerability and network properties. If node degree information is guaranteed not to be known to the attacker, then our framework instantiated with a different attack model could identify different graph metrics that expose node identities. If there are multiple attack models, our framework can be used to infer more sophisticated causal relationships between the graph vulnerability and relevant graph metrics. This feedback can also be used to compare the strengths of different attack models. We think this is a promising future work direction to which our framework can contribute significantly. However, we empirically proved that for such an attack, assortativity and transitivity are revealing much information about node identities. Finally, this study could be extended to understand the intrinsic vulnerability of dynamic graphs, or graphs with node and edge attributes.

Our framework fills a gap between theoretical research and practice, and provides a unifying platform for the development of new methodologies related to graph anonymization, de-anonymization and graph vulnerability quantification. Specifically, this framework can be used to select the particular tradeoff between acceptable vulnerability and needed utility in terms of graph metrics. Data owners should carefully design anonymization algorithms given the require-

ment of privacy and utility with respect to the quantified tradeoff identified by our framework. They would re-evaluate or re-design the anonymization algorithm with such feedback. Alternatively, this framework can be used to empirically calibrate theoretical estimations of privacy, such as techniques based on differential privacy. In a different context, this framework could be used to inform a network alignment problem about the possible conditions for a perfect matching.

Chapter 3: Privacy of Labeled Networks[2]

As shown in the previous chapter, we identify which structural properties contribute most to graph vulnerability. However, in practice, most networks have node attributes such as labels that identify nodes as cheaters or noncheaters in online gaming platforms [89]. The effects of node attributes on the risks of node re-identifications are not yet well understood. While intuitively any extra piece of information can be a danger to privacy, a rigorous understanding of what topological and attribute properties affect the re-identification risks is needed. In cases such as information dissemination, node attributes may be informed by the local graph topology. *How does the interplay between topology and node attributes affect node privacy?*

This chapter assesses the additional vulnerability to re-identification attacks posed by the attributes of a labeled graph. We consider exactly one binary attribute to understand the lower bound of the damage that node attributes inflict. We focus our empirical study on the interplay between topology and labeling as a leverage point for re-identification. While most efforts for re-identification attacks are meant to show the vulnerability or resilience of a particular anonymization technique, this work is different, as it focuses on understanding in which conditions node re-identification is feasible, given the network topology and node attributes. Consequently, whether the network topology is original or anonymized is irrelevant for our study. We extend the privacy framework as introduced in the previous chapter for both topological and attribute information to re-identify nodes. Our study involves real-world graphs and synthetic graphs in which we control how labels are placed

[2]This chapter was previously published in [31, 32]. Permission is included in Appendix A.

relative to ties to mimic the ubiquitous phenomena of homophily—the tendency to connect with similar people—found in social graphs [90].

Our empirical results show that the vulnerability to node re-identification depends on the population diversity with respect to the attributes considered [31]. Using information about the distribution of labels in a node's neighborhood provides additional leverage for the re-identification process, even when labels are rudimentary. In this study, we show more evidence on this phenomenon based on the well-studied Susceptible-Infectious (SI) epidemic model. Furthermore, we quantify the relative importance of attribute-related and topological features in graphs of different characteristics.

The remainder of this chapter is organized as follows. Section 3.1 outlines the related work. The improved privacy framework is presented in Section 3.2. Section 3.3 describes the characteristics of the datasets we used in our empirical investigations. We present our results in Section 3.4 and discuss our contributions in Section 3.5.

3.1 Related Work

Recently, there have been efforts to incorporate node attribute information into deanonymization attacks. Gong et al. [91] evaluate the combination of structural and attribute information on link prediction models. Attributes not present may be inferred through prior knowledge and network homophily. Qian et al. [92] apply link prediction and attribute inference to deanonymization by quantifying the prior background information of an attacker using knowledge graphs. In knowledge graphs, edges not only represent links between nodes but also node-attribute links and link relationships among attributes. The deanonymization attack in [93] maps node-attribute links between an anonymized graph and its auxiliary. In addition to structural similarity, nodes are matched by

attribute difference, the union of the attributes of the node in the anonymized and auxiliary divided by their intersection.

Several researchers propose theoretical frameworks to examine how vulnerable or deanonymizable any (anonymized) graph dataset is, given its structure [52, 56, 55, 45]. However, some techniques are based on unrealistic data models (e.g., Erdös-Rènyi (ER) models [52]), while others make impractical assumptions about the seed knowledge [55]. Ji et al. [45] also introduced a configuration model to quantify the deanonymizablity of graph datasets by considering the topological importance of nodes. The same set of authors analyzed the impact of attributes on graph data anonymity [93]. They show a significant loss of anonymity when more node-attribute relations are shared between anonymized and auxiliary graph data. Specifically, they measure the entropy present in node-attribute mappings available for an attacker. As the entropy decreases, the graph loses node anonymity.

The main aspects distinguishing this study from existing works are as follows: i) In our work, we study the inherent conditions in graphs that provide resistance/vulnerability to a general node re-identification attack based on machine learning techniques. ii) To the best of our knowledge, this is the first work that quantifies the privacy impact of node attributes under an attribute attachment model biased towards homophily. iii) We analyze the interplay between the intrinsic vulnerability of the graph structure and attribute information.

3.2 Modeling Privacy Based on Network Properties and Node Labels

Our main objective is to quantitatively estimate the vulnerability to re-identification attacks added by node attributes. In particular, we ask: *Given a graph topology, how much better does a*

node re-identification attack perform when the node attributes are included in the attack compared to when there is no node attribute information available to the attacker?

We are interested in measuring the intrinsic vulnerability of a graph with attributes on nodes, in the absence of any particular anonymization technique on topology or node attributes. The intuition is that particular graphs are inherently more private: for example, in a regular graph, nodes are structurally indistinguishable. Adding attributes to nodes, however, may contribute extra information that could make the re-identification attack more successful. Consider another example, in a highly disassortative network (such as a sexual relationships network), knowing the attribute values (i.e., gender) of a few nodes will quickly lead to correctly inferring the attribute values of the majority of nodes, and thus possibly contributing to the re-identification of more nodes. Thus, we also ask the following question in this study: *How does the distribution of node attributes affect the intrinsic vulnerability to a re-identification attack of a labeled graph topology?*

To answer these question, we improved the machine learning-based re-identification attack model from our previous work [30]. We use a similar threat model as before that aims at finding a bijective mapping between nodes in two different labeled graphs (Section 3.2.1). We mount a machine-learning based attack by employing additional node attribute features, in which the algorithm learns the correct mapping between some pairs of nodes from the two graphs, and estimates the mapping of the rest of the dataset (Section 3.2.2).

3.2.1 The Attack Model

The threat model we consider is the classical threat model in this context [52]. The attacker aims to match nodes from two networks whose edge sets are correlated. We assume each node

is associated with a binary valued attribute, and this attribute is publicly available. Common examples of such attributes are gender, professional level (i.e., junior or senior), or education level (i.e., higher education or not).

For clarity, consider the following example: an attacker has access to two networks of individuals in an organization that represent the communication patterns (e.g., email) and friendship information available from an online social network. Individuals in the communication network are described by professional seniority (e.g., junior or senior), while individuals in the friendship network are described by gender. These graphs are structurally overlapping, in that some individuals are present in both graphs, even if their identities have been removed. The attacker's task is to find a bijective (i.e., one-to-one) mapping between the two subsets of nodes in the two graphs that correspond to the individuals present in both networks.

We assume that the adversary has a sanitized graph G_{san} that could be associated with an auxiliary graph G_{aux} for the re-identification attack (as depicted in Figure 3.1). As in the scenario discussed above, G_{san} could be the communication network, while G_{aux} is the friendship network of a set of individuals in an organization. We use the same algorithm as presented in Section 2.2.2 to find the bijective mapping between G_{san} and G_{aux}.

3.2.2 Topology and Node Labels

Since we are employing machine learning techniques to re-identify nodes in a graph, we need to represent nodes as feature vectors. We define the node u's features using a combination of two vectors made up from its neighborhood degree distribution (NDD) (as explained in the Section 2.2.2.2) and neighborhood attribute distribution (NAD) (as depicted in Figure 3.2).

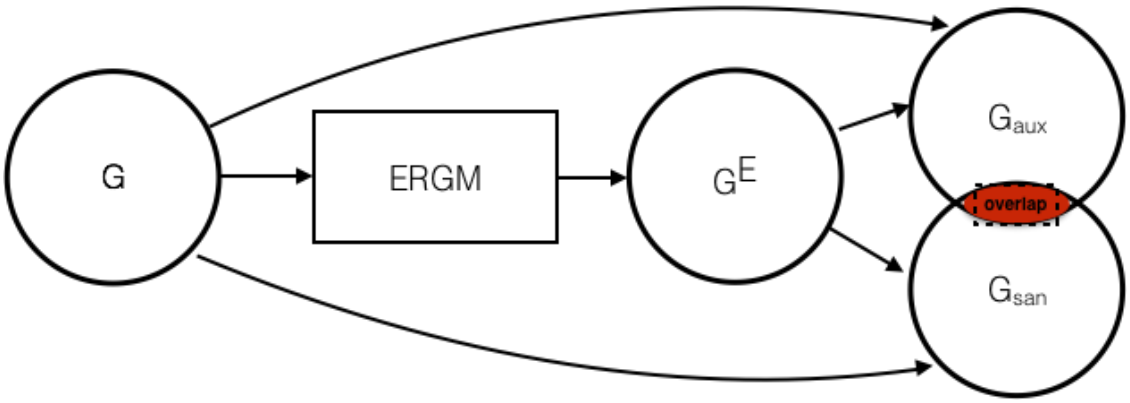


Figure 3.1: The overview of generating identical and non-identical node pairs. The nodes in the overlap are identical to both G_{aux} and G_{san}, but they have different structural characteristics.

NAD is defined by $NAD_u^q[i]$ which represents the number of u's neighbors at distance q with an attribute value i. It was shown experimentally that the use of neighbor attributes as features often improves the accuracy of edge classification tasks [94].

We use the notation GS to represent the prediction results from the input features made up from the topology (e.g., NDD). GS(LBL) to represent features from both the topology and attribute information (e.g., concatenation of NDD and NAD vectors).

Note that the nodes in $G_{san} \cap G_{aux}$, common to both graphs, can be recognized as being the same node (identical) in the two graphs based on their node identifier. Non-identical nodes are unique to each G_{san} and G_{aux} and would not exist in the overlap. In the classification task, we wish to output 1 for an identical node pair and 0 for a non-identical node pair. This is the ground truth against which we measure the accuracy of the learning algorithms. We generate examples for the training phase of the deanonymization attack by randomly picking node pairs from the sanitized (G_{san}) and the auxiliary (G_{aux}) graphs, respectively.

As described previously in Section 2.2.2, we train a classifier to differentiate two nodes as identical or not. For each graph, we take $\ell = 1000$ balanced sub-samples randomly and perform

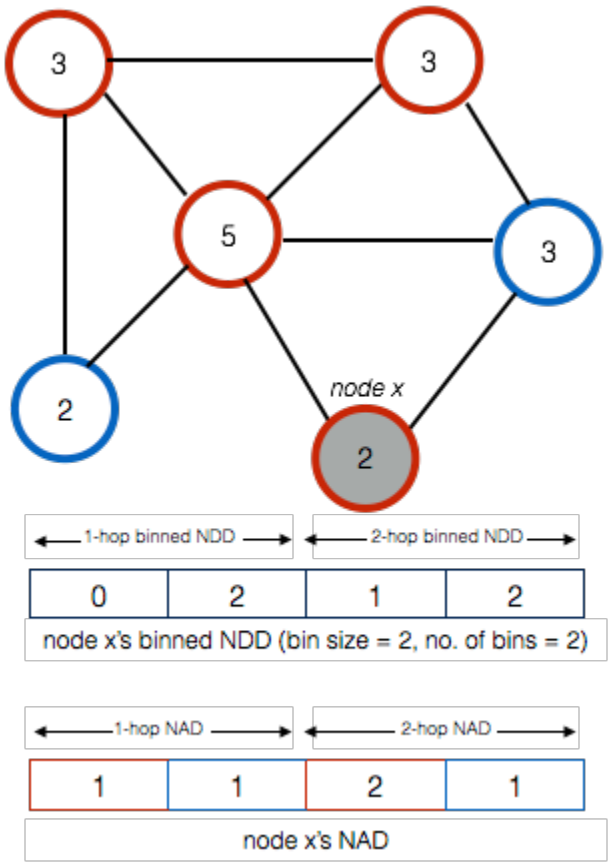


Figure 3.2: Example feature vector made up from NDD and NAD vectors. In the NAD vector, each element corresponds to the number of nodes with the given attribute. Both 1-hop and 2-hop NADs are calculated and merged. Node x has one 1-hop neighbor node, and two 2-hop neighbor nodes with the attribute *Red*. Note that the node value represents the associated degree, and the border color represents the node attribute *Red* or *Blue*.

5×2 cross-validation to evaluate the classifier using the mean F1-score. We thus build two vectors of mean F1-scores, each of size $\ell = 1000$, one for the labeled (GS(LBL)) and one for the unlabeled network topology (GS). An important aspect of these vectors is that they are related in the sense that the i^{th} element in one vector represents the same sample as the i^{th} element of the other vector. This is important for the pairwise comparison of the two mean F1-score vectors.

We perform a standard T-test on these two vectors and report the T-statistic value. The T-statistic value is a measure of how close to the hypothesis an estimated value is. In our case,

the hypothesis is the prediction accuracy of the node identities in the unlabeled graph (GS) and the estimated value is the prediction accuracy in the labeled graph (GS(LBL)). Thus, a large T-statistic value implies a significantly better prediction accuracy of node identities in GS(LBL) than in GS. In such cases, we can say that the network with node attributes is more vulnerable to node re-identification. This value serves as our statistical measurement to quantify the vulnerability cost of node attributes.

3.3 Datasets

Because our work is empirically driven, a larger set of test datasets promises a better understanding of the relations between vulnerability to re-identification attacks and the particular characteristics of the node attributes (such as fractions of attributes of a particular value or the assignment of attributes to topologically related nodes). In this respect, real datasets are always preferable to synthetic ones, as they potentially encapsulate phenomena that are missing in the graph generative models. As an example, until very recently, the relation between the local degree assortativity coefficient and node degree was not captured in graph topology generators [87].

However, relying only on real datasets has its limitations, due to the scarcity of relevant data (in this case, networks with binary node attributes) and the difficulty of covering the relevant space of graph metrics when relying only on available real datasets. Thus, in this work, we combine real networks (described in Section 3.3.1) with synthetic networks generated from the real datasets. For generating synthetic labelled networks, we employ ERGMs [81, 82] and a controlled node-labeling algorithm as described in Section 3.3.2.

3.3.1 Real World Networks

We chose six publicly available datasets from four different contexts and generated eight networks with binary node attributes.

- *polblogs* [95] is an interaction network between political blogs during the lead up to the 2004 US presidential election. This dataset includes ground-truth labels identifying each blog as either conservative or liberal.
- *fb-dartmouth*, *fb-michigan*, and *fb-caltech* [96] are Facebook social networks extant at three US universities in 2005. A number of node attributes such as dorm, gender, graduation year, and academic major are available. We chose two such attributes that could be represented as binary attributes: gender and occupation, whereby occupation we could identify the attribute values "student" and "faculty". From each dataset, we obtained two networks with the same topology but different node attribute distributions.
- *pokec-1* [97] is a sample of an online social network in Slovakia. While the Facebook samples are university networks, Pokec is a general social platform whose membership comprises 30% of the Slovakian population. *pokec-1* is a one-fortieth sample. This dataset has gender information available as a node attribute.
- *amazon-products* [98] is a bi-modal projection of categories in an Amazon product co-purchase network. Nodes are labeled as "book" or "music", edges signify that the two items were purchased together.

As Table 3.1 shows, the networks generated from these datasets have different graph characteristics. For example, the density (d) of the graphs varies across three orders of magnitude,

Table 3.1: Graph properties of the real network datasets. All graphs are undirected, and nodes are annotated with a binary valued attribute. E.g., nodes in the polblogs network have the attribute *party* with values; conservative and liberal. For simplicity, binary values are presented using the notation of R and B, together with the distributions of such values over nodes and edges. p and τ present the estimated parameter values of the attraction model. Density ($\bar{d}$) is the fraction of all possible edges, transitivity (C) is the fraction of triangles of all possible triangles in the network. degree-assortativity (r) measures the similarity of relations depending on the associated node degree. Average path length (κ) depicts the average shortest path length between any pairs of nodes.

Network	Number of nodes		Number of edges			p	τ	$\bar{d}$	C	r	κ
	$R(\%)$	$B(\%)$	$R-R(\%)$	$B-B(\%)$	$R-B(\%)$						
polblogs	1224		16718					0.02	0.22	−0.22	2.49
(party)	48	52	44	48	8	0.48	0.84				
fb-caltech	769		16656					0.05	0.29	−0.06	1.33
(gender)	91.5	8.5	92.8	0.2	7	0.08	0.52				
(occupation)	72	28	69	8	23	0.28	0.42				
fb-dartmouth	7694		304076					0.01	0.15	0.04	2.76
(gender)	86.5	13.5	83.2	0.9	15.9	0.14	0.34				
(occupation)	62	38	58	18	24	0.38	0.5				
fb-michigan	30147		1176516					0.0026	0.13	0.115	3.05
(gender)	92.2	7.8	90.5	0.2	9.3	0.08	0.37				
(occupation)	77.5	22.5	72	9	19	0.22	0.46				
pokec-1	265388		700352			0.46	0	2×10^{-5}	0.0068	−0.044	5.66
(gender)	46	54	18.6	22.4	59						
amazon-products	303551		835326			0.18	0.99	1.8×10^{-5}	0.21	−0.06	17.42
(category)	82	18	83.4	16.4	0.2						

while degree assortativity oscillates between disassortative (for *polblogs*, $r = -0.22$, where there are more interactions between popular and obscure blogs than expected by chance) to assortative (as expected for social networks). All topologies except for *amazon-products* have small average path length.

This wide variation in graph metrics values is what motivated our choice for these set of real networks. We opted to include the three Facebook networks from similar contexts to also capture more subtle variations in network characteristics.

3.3.2 Synthetic Networks

In order to be able to control graph characteristics and node attribute distributions, we also generated a number of synthetic graphs comparable with the real datasets just described. The graph generation included two aspects: topology generation, for which we opted for ERGMs, and node attribute assignments, for which we implemented the technique proposed in [99]. We discussed Exponential-family random graph models (ERGMs) in the previous chapter (Section 2.3.2.2). As before, we generate ERGM synthetic networks based on clustering coefficient (*ERGM-cc*), average path length (*ERGM-apl*), and assortativity (*ERGM-dc*) of the original networks using the R [83] and the *statnet* suite [84].

We use the "attraction" model [99] to generate binary node attributes. This model parameterizes a labeled graph with a tendency towards homophily (ties disproportionately between those of similar attribute background). In the basic case of a binary attribute variable and a constant tendency to inbreed, two parameters, p and τ, both in the (0,1) interval, characterize the distribution of ties within and between the two groups. The first is the proportion of the population that takes on one value of the attribute (with $1 - p$, the proportion taking on the other value). The second parameter, the inbreeding coefficient or probability, expresses the degree to which a tie whose source is in one group is "attracted" to a target in that group. When $\tau = 0$, there is no special attraction and ties within and between groups occur in chance proportions. When $\tau > 0$, ties occur disproportionately within groups, increasing as τ approaches 1. Given a total number of ties, values for p and τ determine the number of ties/edges that are between groups, namely, $\delta = |E| \times 2 \times (1 - \tau)p(1 - p)$. Intuitively, p captures the diversity of attribute values in the node population (with $p = 0.5$ showing equal representation of the attributes) while τ captures the ho-

mophily phenomenon (that functions as an attraction force between nodes with identical attribute values).

We report the p and τ values for the original network as shown in Table 3.1. The homophilic attraction metric τ varies between 0 in *pokec-1* (thus, no higher than chance preference for social ties with people of the same gender in Slovakia) to 0.99 in *amazon-products* (books are purchased together with other books much more strongly than given by chance). The diversity metric p varies between the over representation of males in the US academic Facebook networks (8% female representation) to an almost perfect political representation in the *polblogs* dataset (where $p = 0.48$). Note that, we only consider p as the minimum proportion of two node groups due to the symmetric nature of attributes in our experiments.

In the process of generating synthetic node attributes, we first randomly assign two arbitrary values (i.e., R and B) as labels to all the nodes in the graph for a given $p, 1-p$ split. Then, we draw an R node and a B node at random and swap labels if it would decrease the number of R-B ties. This process would converge when the total number of cross-group ties reduce to δ for a particular value of τ.

Figure 3.3 shows the proportion of cross-group ties on the synthetic labelled networks generated from *polblogs* topology. The proportion of cross-group ties is proportional to p, while it is inversely proportional to τ. When p reaches its maximum ($p_{max} = 0.5$ due to the symmetric nature of binary attribute values), the proportion of cross-group ties is larger at minimum inbreeding coefficient τ. It should be noted that convergence is not guaranteed for all possible combinations of p and τ. The swapping procedure holds constant all graph properties except the mapping of nodes to labels, and consequently, it may not be possible to find a mapping of nodes to labels that achieves

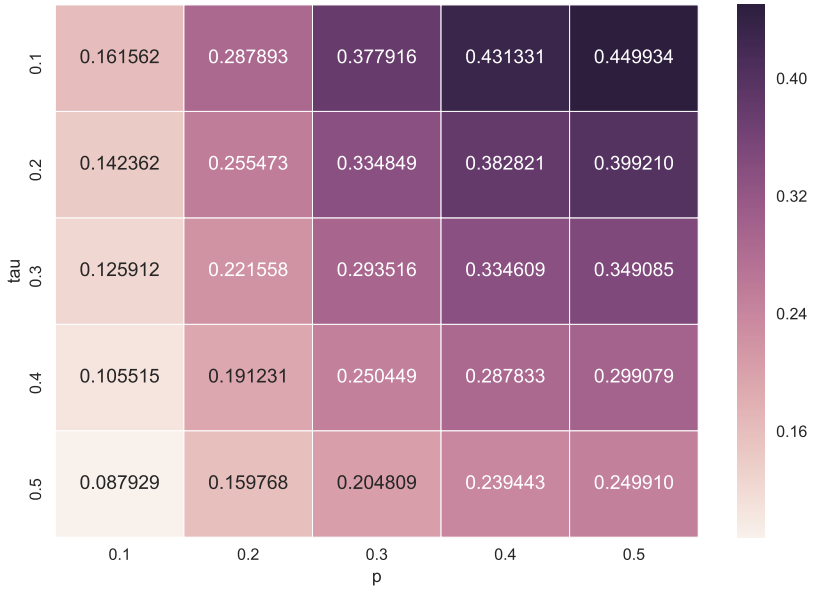


Figure 3.3: Proportion of cross group ties. We report the proportion of cross group ties on synthetic labeled networks generated from the polblogs network. We use the "attraction" model [99] to generate binary node attributes.

a target number of ties between groups (when that number is low as it is for higher values of τ). Table 3.2 presents the graph characteristics of the synthetically generated labeled graphs.

3.4 Empirical Results

Our objective is not to measure the success of re-identification attacks on original datasets in which node identities have been removed: it has been demonstrated long ago [48] that naive anonymization of graph datasets does not provide privacy. Instead, our objective is to quantify the exposure provided by node attributes on top of the intrinsic vulnerability of the particular graph topology under attack. In our experiments, we leverage the real and synthetic networks described above. We use the methodology described in Section 3.2 to re-identify nodes using features based

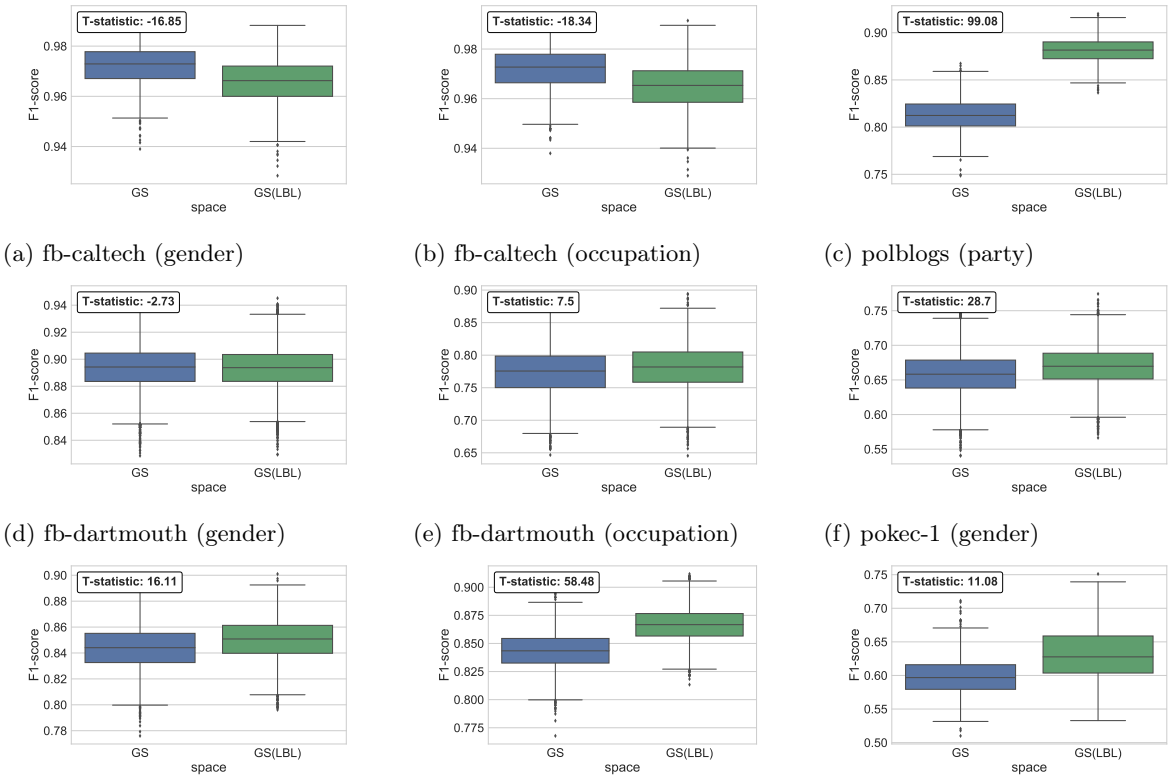
Table 3.2: Basic statistics of generated ERGM networks. Note that *dc, cc* and *apl* define the set of parameters used to generate ERGM graphs based on assortativity (degree correlation), clustering coefficient, and average path length, respectively. We generated a total of ≈ 500 million identical and non-identical node pairs over three ERGM graph spaces of the six real social network datasets. S is the population of generated node pairs concerning a given graph topology.

Network	ERGM	d	C	r	κ	$\|S\|$ (*millions*)
polblogs	dc	0.02	0.03	.08	2.52	5.5
	cc	0.02	0.33	-0.02	2.69	13.1
	apl	0.02	0.10	-0.06	2.49	11.5
fb-caltech	dc	0.06	0.08	0.11	2.13	1.2
	cc	0.06	0.42	-0.06	2.73	4.1
	apl	0.06	0.07	0.11	1.97	1.2
fb-dartmouth	dc	0.01	0.17	0.07	2.66	14.5
	cc	0.01	0.24	0.04	2.77	13.2
	apl	0.01	0.20	0.04	2.70	14.2
fb-michigan	dc	0.003	0.02	0.12	3.28	38.4
	cc	0.002	0.20	0.12	3.52	39.9
	apl	0.002	0.20	0.12	3.64	38.2
pokec-1	dc	2.02E-5	0.06	-0.04	5.60	29.5
	cc	2.05E-5	0.07	-0.04	5.84	29.3
	apl	2.04E-5	0.06	-0.04	5.63	27.3
amazon-products	dc	1.82E-5	0.37	-0.06	11.86	43.7
	cc	1.82E-5	0.40	-0.06	13.52	72.5
	apl	1.82E-5	0.39	-0.06	13.47	74.3

on both graph topology and node attributes. Our first guiding question is thus: *How much risk of node re-identification is added to a network dataset by its binary node attributes?*

3.4.1 The Vulnerability Cost of Node Attributes

Figure 3.4 presents the accuracy of node re-identification in the original graph topology GS and in the same topology augmented with node attributes GS(LBL). As expected, the re-identification attack performs (generally) better when node attributes are used in the attack. Surprising to us, however, is the relatively small vulnerability cost that node attributes introduce. For example, the *occupation* attribute has a barely noticeable benefit to the attacker in *fb-dartmouth*. More interestingly, however, the same attribute performs differently for the other two Facebook net-



(a) fb-caltech (gender)

(b) fb-caltech (occupation)

(c) polblogs (party)

(d) fb-dartmouth (gender)

(e) fb-dartmouth (occupation)

(f) pokec-1 (gender)

(g) fb-michigan (gender)

(h) fb-michigan (occupation)

(i) amazon-products (category)

Figure 3.4: Accuracy of predictions over original networks. Mean accuracy values are shown for real network datasets on GS and GS(LBL) along with the T-statistic which describes the difference in means of the GS and GS(LBL) vectors of prediction probabilities statistically. The network with node attributes is more vulnerable to node re-identification when the T-statistic is positive and large.

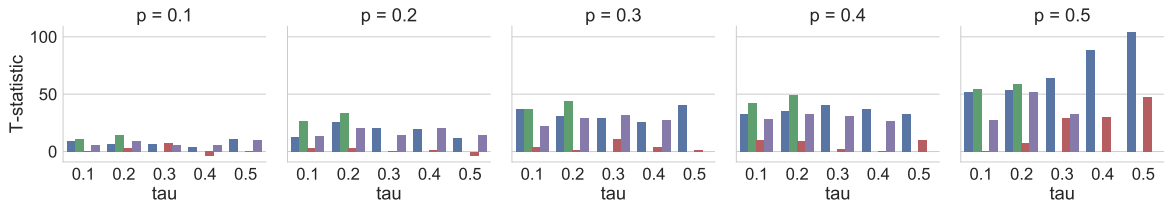
works considered: for *fb-caltech* the *occupation* label functions as noise, leading to a small decrease in the F1-score. For *fb-michigan*, on the other hand, the *occupation* label significantly improves the attacker's performance.

Another observation from this figure is that different node attributes applied to the same topology have different outcomes: see, for example, the case of the *fb-michigan* topology, where the difference between the impacts of the *gender* and the *occupation* attributes is the largest. We thus formulate a new question: *What placement of attributes onto nodes reveal more information?* To un-

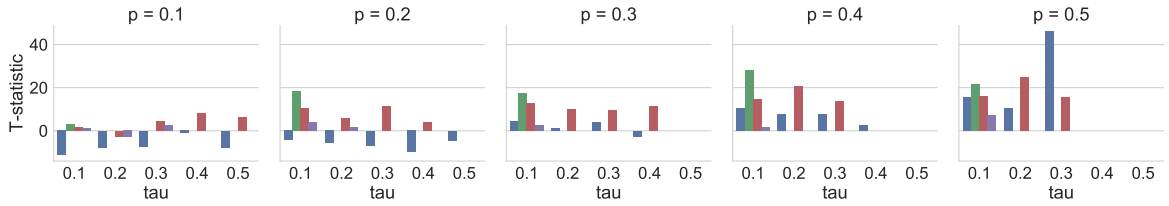
derstand how the placement of attribute values on nodes affects vulnerability, we generate synthetic node attributes in a controlled manner by varying p (the diversity ratio) and τ (homophily coefficient). This allows us to study the effect of these parameters on node re-identification. Figure 3.5 presents the T-statistics of the F1-scores for node re-identification attacks on the original topology vs. labeled versions of the original topology. In addition to the original topologies, Figure 3.5 also presents results on various synthetic networks generated as presented in Section 3.3.2.

We observe three phenomena: First, it appears that p is positively correlated with the T-statistic value measuring the re-identification impact of attributes. That is, the more diversity (that is, the larger p), the more vulnerable to re-identification the labeled nodes become on average. Intuitively, in a highly skewed attribute population, while the minority nodes will be identified quicker due to node attributes, the majority remains protected. On the other hand, when $p = 0.5$, a network has two equal-sized sets of nodes where each set takes one of two attribute values. This is explained by the fact that the NAD feature vector captures more diverse information in the attributes of neighbots when p is larger. This is also the explanation for why the node attributes contribute so much more to vulnerability in the *polblogs* dataset, which has a large diversity ($p = 0.48$) (thus, almost equal numbers of conservative and liberal blogs). Note that the effect of p on the added vulnerability remains consistent across all topologies (real and synthetic) tested.

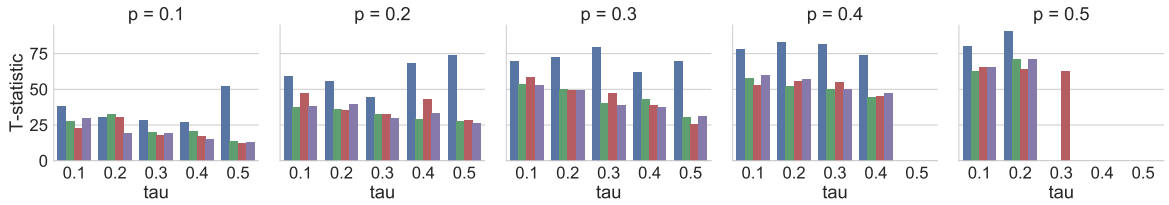
The second observation is that there is no visible pattern on how the inbreeding coefficient (τ) influences the vulnerability added by binary node attributes. While this is disappointing from the perspective of story telling, it is potentially encouraging for data sharing, as it suggests that datasets that record homophily (or influence, the debate is irrelevant in this context) do not have to be anonymized by damaging this pattern. For example, the privacy of a dataset that records an



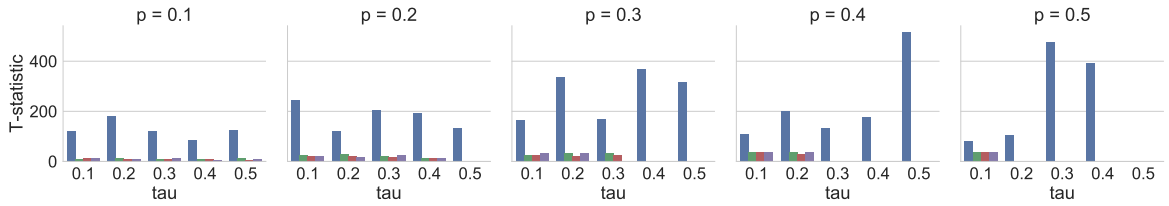
(a) polblogs



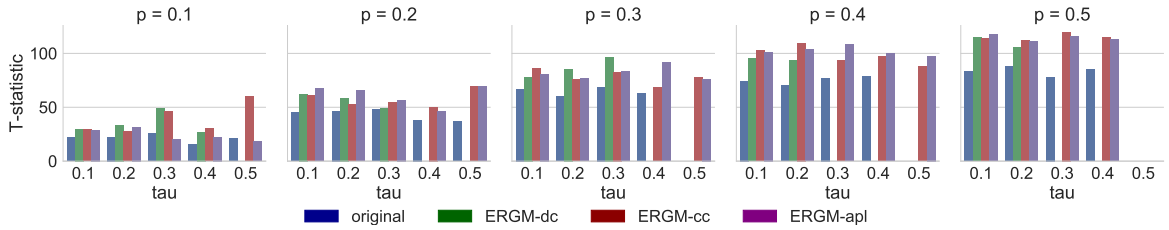
(b) Network: fb-caltech



(c) fb-michigan



(d) pokec-1



(e) amazon-products

Figure 3.5: T-statistic between prediction scores of GS(LBL) and GS networks. GS represents the network structure and GS(LBL) represents the networks with varying attributes. Results are shown across different structures of original and ERGM graphs. Each ERGM graph is presented using the generated parameters of *dc* (degree-correlation), *cc*(clustering coefficient) and *apl* (average path length). We skip presenting *fb-dartmouth* in this figure to reduce visual clutter.

information dissemination phenomenon could be provided without perturbing the cascading-related ties.

The third class of observations is related to the relative effect of the topological characteristics on the added vulnerability. Both *amazon-products* and *pokec-1* are orders of magnitude sparser than the other datasets considered. This means that the topological information available to the machine learning algorithm is limited. In this situation, the addition of the attribute information turns out to be very significant: the T-statistic values for these datasets are significantly larger than for the other datasets, with values over 400 in some cases. Another topological effect is noticed when comparing the real *pokec-1* topology with the ERGM-generated ones in Figure 3.5d: the node attribute contributes much more to the vulnerability of the original topology compared to the synthetic topologies. The reason for this unusual behavior may lay in the different clustering coefficients of the networks, as seen in Tables 3.1 and 3.2: the ERGM-generated topologies have clustering coefficients one order of magnitude higher than the original topology (for the same graph density), which leads to more diverse NDD feature vectors for the networks with higher clustering and thus richer training information. This in turn leads to better accuracy in node re-identification in the unlabeled ERGM topologies (with higher clustering) than in the original topology. For example, the maximum F1-score for the ERGM-dc topology is 0.92 while for the original it is 0.76 in *pokec-1*. Thus, the relative benefit of the node attribute was significantly higher when the topology features were poorer.

3.4.2 The Impact of Topology

Figure 3.6 presents the importance of features that are used in node re-identification. A high importance score represents a feature that is responsible for accurately classifying a large proportion

of examples. We make three observations from this figure. First, most of the NAD features that represent node attribute information prove to be important in all datasets.

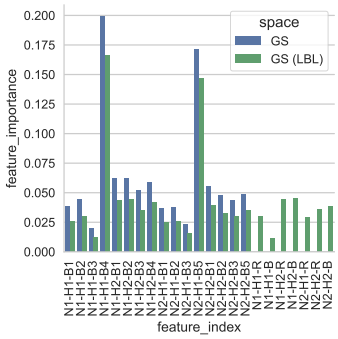
Second, among the NDD features, only a small number contributes consistently to accurate prediction. As shown in Figures 3.6c - 3.6i, the first bin of 1-hop and 2-hop NDD vectors contribute the most. That is, a high impact on the re-identification of a node is brought by the number of its neighbors with degrees between 1 and 50. Even in large networks such as *pokec-1* and *amazon-products* with a larger range of node degrees, this behavior is observed.

Third, Figure 3.6 suggests what features explain the effect of diversity p on node re-identification in labeled networks. On datasets with large diversity (such as *polblogs* or *pokec-1*), the topological information contributes less than on datasets with low diversity (such as *fb-caltech (gender)*). This is because high diversity correlates to richer NAD feature vectors, and thus the relative importance of the NAD features increases.

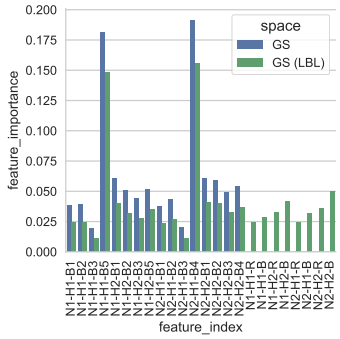
3.4.3 Epidemic and the Risk of Node Re-identification

In this section, we consider the scenario of node attribute placement under the constraint of an epidemic process. We use the Susceptible-Infectious (SI) [100] model to generate an epidemic process on the original graph topology. In the SI model, individuals are initially susceptible with the exception of a small fraction of the population who is infectious. In contact with an infections individual, a susceptible individual becomes infectious with the probability β. Once infected, individuals stay infected and infectious throughout their lifetime.

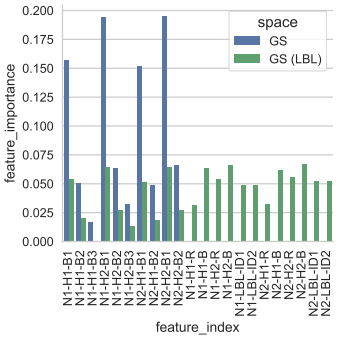
We use this model to assign binary attributes (i.e., susceptible and infectious) to the nodes in the graph. In each experiment, we select the 0.1% highest degree nodes as infectious to initialize



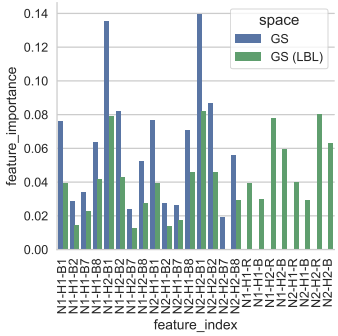
(a) fb-caltech (gender)



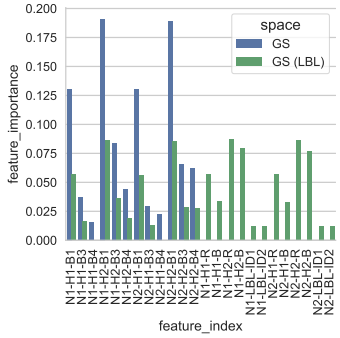
(b) fb-caltech (occupation)



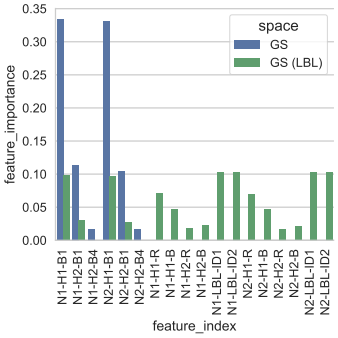
(c) polblogs (party)



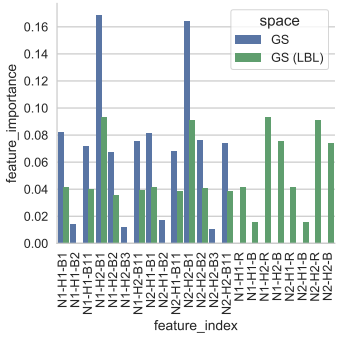
(d) fb-dartmouth (gender)



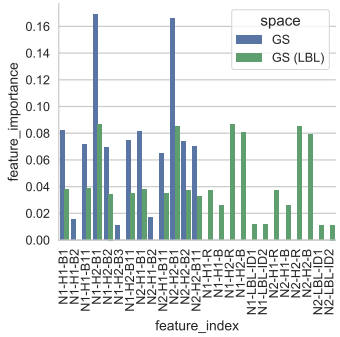
(e) fb-dartmouth (occupation)



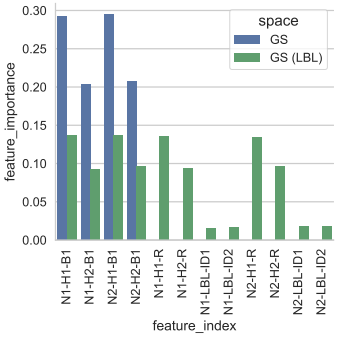
(f) pokec-1 (gender)



(g) fb-michigan (gender)



(h) fb-michigan (occupation)



(i) amazon-products (category)

Figure 3.6: The importance of features across original networks. NDD features are presented in the index order of node (N), hop (H) and bin (B). As an example, the feature *N1-H2-B1* presents the first bin of the $NDD_1^2[k]$ vector. NAD features are presented in the index order of node (N), hop (H) and binary attribute value $\in R, B$. As an example, the feature *N1-H2-R* presents $NAD_1^2[R]$. Any feature that does not contribute to the final prediction decision with at least 1% of the samples in average is omitted.

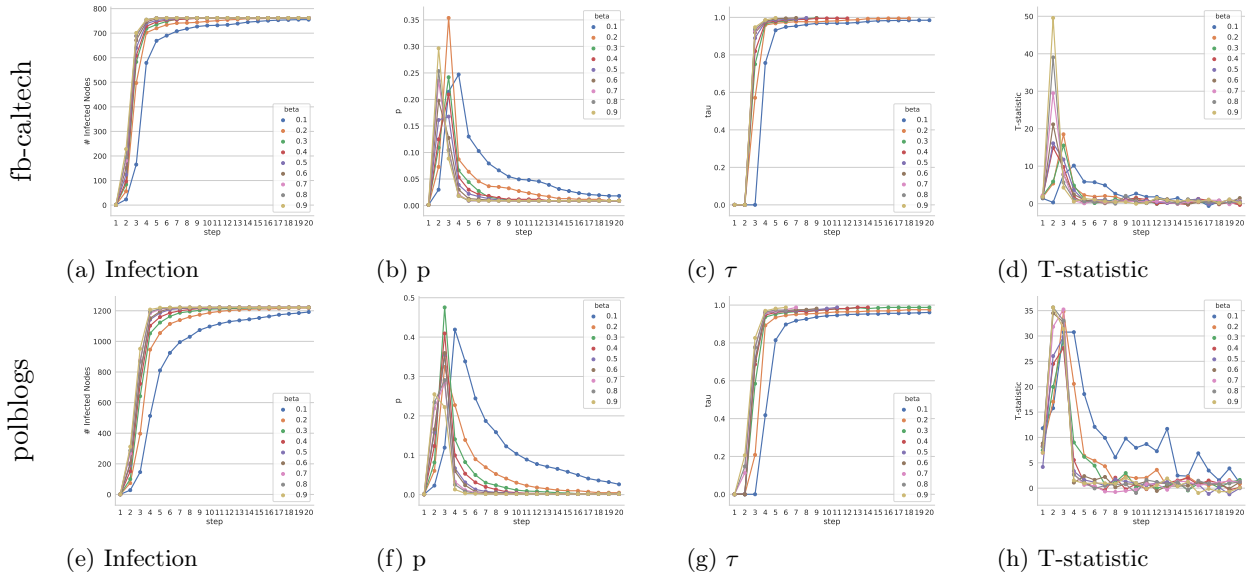
the epidemic. We vary the infection probability β between 0 and 1. We mount the machine-learning attack to each epidemic graph independently on the graph topology GS and on the same topology augmented with binary node attributes GS(LBL). We make two assumptions in this task. First, we assume that the graph topology remains static during the epidemic process. Second, we assume that the adversary does not have any prior information about other epidemic graphs in the series.

We calculate the significance of the vulnerability scores in GS(LBL) compared with GS via a standard T-test, and report the T-statistic value per each epidemic graph. Figure 3.7 shows the T-statistic values over multiple steps in the epidemic process including other characteristics (e.g., the node infection probability β, the estimated homophily τ observed in the network).

We observe the same phenomena on the correlation between population's diversity (p) and the T-statistic values over the epidemic graphs. However, the T-statistic values show different patterns depending on the infection probability β. Note that, the population's diversity (p) increases to a local maximum in the initial time-steps, and then drops in later time-steps. This is an intuitive observation given the properties of SI model [100].

When the epidemic grows slowly (i.e., low infection probability), the T-statistic value also increases at a slower rate. On the other hand, when the epidemic outbreaks at a faster infection rate, the T-statistic value also increases at a higher rate and achieves a relatively larger peak value. For the *fb-caltech* network, the T-statistic value reaches a peak value of 10 in four infection steps for $\beta = 0.1$, while the T-statistic value reaches a peak value of 50 in two infection steps for $\beta = 0.9$. Interestingly, the most diverse population in *fb-caltech* network is also observed after four infection steps for $\beta = 0.1$, and two infection steps for $\beta = 0.9$ (as shown in Figure 3.7d). In *polblogs*, T-statistic values reach peak values of 31 and 36 for the infection rates of 0.1 and 0.9, respectively

(as shown in Figure 3.7h). The *polblogs* population becomes more diverse in the similar number of infection steps given the respective infection rate.



(a) Infection (b) p (c) τ (d) T-statistic

(e) Infection (f) p (g) τ (h) T-statistic

Figure 3.7: Graph vulnerability over a series of epidemic graphs under SI model. During each infection step, a susceptible node becomes infected with probability β if in contact with an infected node. (a,e) the rate of infection; (b,f) p – diversity ratio which measures the proportion of nodes with one binary attribute value; (c,g) τ – estimated value of homophily observed in the network; and (d,h) the T-statistic value between prediction scores of GS(LBL) and GS. T-statistic values show how the extra vulnerability due to binary attributes changes over iterations of the epidemic process.

3.5 Summary and Discussion

This chapter shows that the addition of even a single binary attribute to nodes in a network increases the vulnerability to node re-identification. The increase in vulnerability derives from the fact that the machine learning attack makes use of the relationship between topology and the distribution of node labels. Using information about the distribution of labels in a node's neighborhood provides additional leverage for the re-identification process, even when the labels are rudimentary.

Furthermore, we find that a population's diversity with regard to the binary attribute consistently degrades anonymity and increases vulnerability. Diversity means a more even distribution of the binary attribute, which produces a more varied set of neighborhood distributions that nodes can exhibit. Consequently, nodes are more easily distinguished from one another by virtue of their differing neighborhood distributions of labels.

This observation is critical for network datasets for which the node attributes are the result of an epidemic process. If the epidemic process is monitored, an adversary could observe the node states and their changes repeatedly over multiple time steps. In such a scenario, the adversary could mount a strong node re-identification attack. The techniques presented in this chapter can be applied to build strong anonymization techniques for such cases. Specifically, our techniques can be used to estimate the rate of anonymity loss over the lifespan of an epidemic process and more efficiently guide data owners in the process of network data anonymization.

Another outcome of this work is that there is no consistent discernible impact of homophily, as measured by the inbreeding coefficient, on vulnerability. Our procedure for investigating the impact of homophily simply involves swapping labels without disturbing ties. Therefore, both local and global (unlabeled) topologies remain constant as we decrease the number of cross-group ties to achieve a target value implied by a particular inbreeding coefficient for a given proportional split along the binary attribute. This procedure disturbs the local labeled topology, but because the machine learning attack uses information from that local topology, it apparently can adapt to the changes and make equally successful predictions regardless of the value of the inbreeding coefficient.

Chapter 4: Simulating Social Media Activity

The second part of this dissertation focuses on predicting social media activity. Our goal is to develop techniques for high-fidelity simulations of information spread in online social environments. A reliable simulator can be useful to foresee the spread of information in many real world scenarios. For example, terrorist groups use "Pump and Dump" schemes to raise funds via artificially promoting a digital currency through social media environments [101]. These groups "pump up" specific digital currencies on Twitter to take advantage of short bursts in prices, before they "dump" the currency for a profit [102]. A simulator that is able to predict spikes in social media activity can be used to regulate the "Pump and Dump" attempts. In another instance, a group of Twitter accounts that support the Venezuelan regime amplified certain topics and hashtags to discredit Juan Guaidó, the opposition leader, after the controversial 2019 Venezuelan national election [4]. These attempts were able to manipulate the trending topics on Twitter to control what can be seen by the international community [103]. We can use a simulator to control the spread of certain topics in future discussions. Such a simulator can also be used to evaluate intervention techniques to encourage engagement (e.g., in the case of health information dissemination) or limit misinformation (e.g., by evaluating how misinformation diffuses if some accounts are prevented from engaging).

However, developing a reliable social simulator is not trivial [104, 105]. As shown in Figure 4.1, social media activity can be described at different granularities. The finest granularity of predictions describes when a social media message is posted, who posts the message, what it is about

(topic), and whether it is in response to another message. The complexity of the problem increases when the granularity of the predictions becomes finer. For example, it is easier to predict the volume of social media discussions in a given time interval (*when*) than predicting *who* would interact with such discussions. However, predicting exactly which user will post or reshare a message is a difficult (if at all possible) task based on only observing platform activity. For example, Bollenbacher et al. [106] argue that predicting microscopic user actions is difficult in long-lived online conversations. This limitation is mainly due to the accumulation of errors in long range simulations. In addition, social media content changes rapidly over time as it is subject to both internal (e.g., opinion leaders) and external (e.g., street violence, natural disaster) influences. Thus, a reliable simulator needs to realistically respond to internal and external stimuli.

One distinctive feature of a simulator is the ability to forecast social media activity in future timesteps without relying on the ground truth in the previous time step. This capability can be thought as generalizing single timestep predictions to hundreds of future timestep predictions. This generalization is also associated with the granularity of predictions that one seeks to achieve. For example, one needs to predict whether a particular user is going to post a message tomorrow, or some day within the next week. In another instance, one can develop a timeseries regression model to predict the volume of activities in the next two weeks without relying on any ground truth information in the testing period.

The reliability of a simulator can be measured by different metrics. For example, timeseries predictions can be evaluated by the error between the prediction and ground truth. The interaction between different users can be evaluated by comparing the structural properties of the user interaction network with the ground truth. The performance of simulators is often compared against a

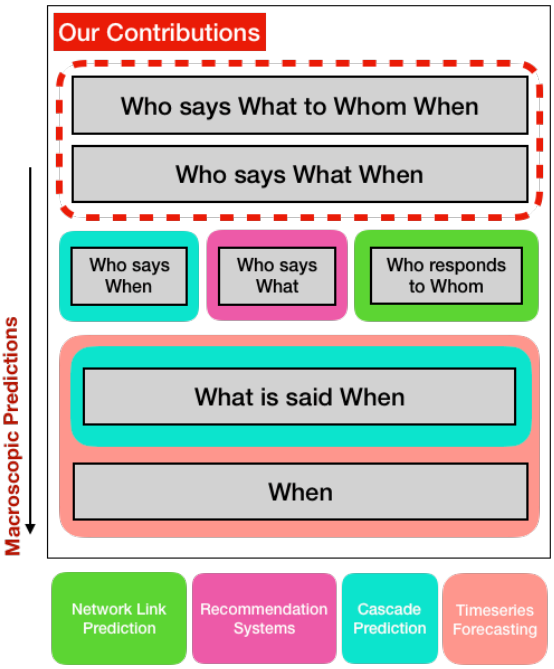


Figure 4.1: The granularity of predictions in decreasing order of complexity. Social media prediction tasks are grouped based on different problem domains addressed in the literature.

baseline model. While there are several ways to create a baseline model for social media simulations, repeating the most recent records in the future is competitive [107].

In this chapter, we first discuss the related attempts that address various parts of the social media activity prediction (Section 4.1). Finally, we explain two scenarios that motivate the design of social simulators presented in this dissertation (Section 4.2). This chapter provides the background for the contributions made in subsequent chapters.

4.1 Related Work

Related work has looked at pieces of the simulation problem using a variety of social media datasets. We group the related work on simulating social media activity into four main problem domains: i) timeseries forecasting, ii) cascade prediction, iii) recommendation systems, and iv)

network link prediction. We map different granularity of social media activity predictions to these problem domains as shown in Figure 4.1.

4.1.1 Timeseries Forecasting

Previous studies developed many regression methods for the timeseries forecasting tasks [108]. Popular statistical methods include Exponential Smoothing (ES) and the Autoregressive Integrated Moving Average (ARIMA). Several deep learning methods such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) have been proposed to deal with both univariate and multivariate timeseries prediction. These generalized methods are applicable to the problem of forecasting the volume of social media discussions over time.

Agent-based-modeling (ABM) techniques use both statistical and deep learning timeseries regression methods to forecast individual user activity streams [107]. For example, Abdelzaher et al. [107] represent each user's activity by a timeseries of K elements, where each element represents the user activity in an arbitrary time slot (e.g., hours, days, etc.). They used both ARIMA and deep neural networks (such as CNN, and RNN) to predict the next K elements of the timeseries. This approach has several limitations. First, they implemented separate models to capture the activity streams of different users. This approach did not scale well when there are millions of users who participate in social media discussions. Second, the majority of users have very few actions, thus making the timeseries very sparse. One way of dealing with this issue is to group users by their activity level. For example, given an activity threshold, we can group users into two sets of active and sparse acting users. While there are rich activity patterns that exist for active users, the actions of inactive users are more likely to be irregular and sparse. In their experiments, Abdelzaher et al. [107] showed competitive performance for sparse acting users when repeating elements of the

past timeseries as predictions of the future. Third, the optimal granularity to define a time slot is different for differently active users. For example, there are hyper active users who share messages frequently within hourly intervals, while others share very few messages within a day [109]. While the granularity of a day seems an optimal choice in general since there are many inactive users in social media environments [110], it would not capture the hourly activity patterns of hyper active users. These smaller granular time predictions are important for scenarios like "Pump and Dump" schemes, where specific groups promote digital currencies on Twitter in short time intervals [102]. A similar challenge is to capture bursty Twitter activity that occur within minutes or hours during exceptional events (e.g., NBA finals) [111].

Graph Convolutional Networks (GCNs) were proposed for univariate timeseries forecasting [112, 113]. One such popular architecture is called Diffusion Convolutional Recurrent Neural Networks (DCRNN). This technique was originally proposed for traffic forecasting and claimed to be general on any univariate timeseries forecasting [112]. Hernandez et al. [109] applied this technique on multiple social media datasets, and showed its poor performance for forecasting user activity timeseries. They provide lessons similar to Abdelzaher et al. [107] as the performance degrades depending on the heterogeneity of user activity. A major understanding is that sparse user activity patterns remain challenging to predict with different forecasting methods.

4.1.2 Cascade Prediction

Many studies tried to predict the popularity of particular social media messages (*what*) over time (*when*) [114]. The popularity prediction task is usually defined as finding the number of reposts that an original message receives in discrete time intervals [115]. Feature-based methods often utilize hashtags, URLs, mentions, sentiment and topics as textual features extracted from

the original messages [116]. Sentiment features are shown to be most useful to predict message popularity [117, 118]. For example, a message becomes more popular when it is associated with negative sentiment [119]. Some works [115, 120, 121] showed that the usefulness of textual features depends on the spread of the content under consideration. In one study [115], content features (e.g., title, caption) are shown to be weak predictors for the popularity of images shared on Facebook. On the other hand, user profile features (e.g., followers, age, etc.) are shown to be more important than content features [120]. Other research suggests that social network structural features (e.g., mean degree, cascade height) and temporal features (e.g., time elapsed, maximum/mean time decay) are effective as user features for the popularity prediction tasks [122]. Several works show the significant advantage of using temporal features in the popularity prediction task [123, 124]. Temporal features are shown to be more useful in smaller cascades than in larger cascades [125].

Many previous works showed that predicting the popularity of a message is not trivial [114]. Later, it was shown that the initial popularity of the message is useful to predict the final state of popularity [126, 127, 106, 128]. Based on this understanding, several methods were developed using deep learning algorithms [126, 15] to predict multiple dimensions of message popularity (e.g., cascade size, shape, virality etc.) given the initial message growth. Embedded-IC [129] embeds cascade nodes in a latent diffusion space to predict the temporal activation of a node. DeepCas [126] proposed a diffusion-embedding framework to predict the incremental growth of a cascade. Both Embedding-IC and DeepCas exploit the paths in a cascade to improve the accuracy of the prediction task. Zayats et al. [130] proposed a graph-structured LSTM model to predict the popularity of Reddit comments in terms of the votes they received. This model was able to distinguish the controversial comments from the positive comments with the help of words associated with humour and emotion categories.

Techniques that predict the popularity of conversations are mostly based on statistical approaches [131, 132, 133]. Wang et al. [134] proposed a theoretical model to capture the temporal evolution of conversation trees by employing a Levy process. They used the preferential attachment mechanism to build conversation trees. Aragon et al. [135] used reciprocity (i.e., strong exchange of messages between users) as a behavioral feature to predict the temporal evolution of a conversation with respect to the depth of a tree. The proposed statistical approach utilizes the mutual dependency between the authorship and conversation structure. Several works [136, 128] model the dynamics of conversation trees using a Hawkes process. Medvedev et al. [136] estimated the parameters from the initial comments of a conversation to predict the remainder. Krohn et al. [128] improved the previous solution in the proposed CTPM model as the parameters are estimated from the post information. More recently, Bollenbacher et al. [106] proposed the Tree Growth Model (TGM) to predict the final size and shape of conversations given the partial conversation tree information. However, the predictive performance of these statistical approaches deteriorate due to the dependence on the chosen parameters and optimization of the likelihood function.

While significant work has focused on predicting individual cascades, less attention has been invested in predicting the popularity of a group of cascades. For example, several works predict the aggregate volume of user activities on Twitter via Hawkes processes that model the events around a group of cascades [137, 138]. Krishnan et al. [139] extracted several structural features from a set of cascade trees (i.e., a forest of cascades) to distinguish viral cascades from broadcasts. Theoretical models that capture the spread of social-influence when a group of competitive cascades evolve over a network have also been proposed [140, 141]. Other works have made similar observations when exploring inter-related cascades in multiplex networks [142]. These studies stress the importance of

focusing on a group of cascades instead of an individual cascade for improving the prediction results of user-level diffusion behavior.

Another line of work develops deep learning models to predict the future actions of social media users. DeepDiffuse [127] is an LSTM architecture to predict the next user who participates in a cascade. Islam et al. [143] used a recurrent neural network architecture to predict a user's next action augmented by her neighbors' recent actions on Flickr, Flixster, Gowalla, and Digg social platforms. TopoLSTM [144] uses the network structure to predict next activated user in a cascade. RBMHDRN [145] was proposed to predict whether a particular user would retweet a given piece of content on Weibo. They extracted a various set of content, user, and network related features to solve this classification task. Myer et al. [146] found that the future action of a user in a cascade is dependent on her previous exposures to multiple other cascades on Twitter. In a similar setting, Weng et al. [147] developed an agent-based model to predict the probability of a user performing a retweet when exposed to multiple memes on Twitter. They discovered an adversely negative and positive effect on simultaneous cascades that are of unrelated and related content, respectively. These solutions are limited to users who have already been seen in the past cascades.

On the other hand, few studies predict the popularity of topics [148, 149], hashtags [150], or keywords [151] shared on Twitter messages. Liu et al. [148] explore machine learning methods to predict whether and when a topic will become prevalent. The authors highlight the challenges faced on forecasting the frequency of topics discussed by users due to irregular patterns. Yin et al. [149] demonstrate that topics prevalent on Twitter can be categorized into temporal topics (e.g., breaking events) and stable topics (e.g., user interests). They utilize both the network structure and temporal information to predict the popularity of temporal and stable topics. Saleiro et al. [151]

classify the popularity of named entity mentions (e.g., "Ronaldo") on Twitter as high or low in the following hours using the features extracted from news articles. They found news articles carry different predictive power based on the nature of the entities under study. Dutta et al. [152] predict the volume of Reddit discussions in a future time horizon leveraging the text from news and an initial set of comments using a recurrent neural network architecture. Shrestha et al. [153] used a deep learning model to forecast the number of retweets and mentions of a specific news source on Twitter using the network structure observed in the day before the predictions. They found that small, but dense network structures are helpful in the predictions.

In summary, many previous studies predicted the growth of cascades in various macro-level properties (e.g., size, virality). They experimented with a variety of features that represent the user, content and temporal attributes. However, many studies assumed to have the initial growth of the cascades as input for this prediction task. This is an impractical assumption to make when simulating social media activity. On the other hand, several studies predicted the popularity of content (e.g., topics, hashtags) spread on social media platforms. While some studies only classified the level of popularity, others assumed to know the ground truth information on the day before predictions. In a simulation task that attempts to predict the social media activity in a future time horizon, we might not have the ground truth information on the previous day of the predictions.

4.1.3 Recommendation Systems

The goal of a recommendation system is to predict the probability of a user to interact with an item [154]. Da'u et al. [155] provide the most recent systematic review on the recommendation system literature. Popular techniques are collaborative filtering (CF) [156, 157, 158], content-based (CB) [159], and deep learning models [160].

CF techniques leverage the user-user and item-item relationships to make recommendation. These techniques suffer from cold start and sparsity issues due to inadequate information present in the both user and item space. For example, when there are new users or new items, CF techniques fail to make any predictions. On the other hand, CB methods are capable of predicting the user who would interact with a new item. In contrast to CF, CB methods use the content description of the item which allows them to make predictions for items that are not seen in the training data. However, CB methods only use the past actions of a user to predict the future, but ignore any related user information. Due to this reason, CB methods are only capable of predicting recommendations for users who are already seen in the training data. While there are many deep learning models such as Autoencoders (AEs), Restricted-Boltzmann-Machines (RBMs), Deep Belief Networks (DBNs), Deep-Boltzmann-Machines (DBMs), Generative Adversarial Network (GAN) proposed in the recommendation systems literature [155], many such methods are evaluated on movie and e-commerce domains, but rarely on social network data.

More recently, Kumar et al. [161] proposed JODIE to predict the user who would interact with a subreddit or Wikipedia page in the future. They used a recurrent neural network model to learn dynamic embedding vectors of users based on a sequence of temporal interactions. GraphRec is another model that used graph neural networks (GNN) to learn the embedding vectors for users present in the user-item graph and the user-user social graph. They represent the edges in the user-item graph with the users' opinions on items (e.g., a user likes item x, and dislikes item y). They found that a combination of social relationship features and opinion features lead to more accurate recommendation results. Both JODIE and GraphRec are not capable of making predictions for users who appear only in the test data.

There is a recent trend of applying machine learning methods to improve social media recommendation algorithms. Most recent attempts in this line of work are due to the annual ACM RecSys grand challenge that is organized by Twitter [162]. In the 2020 challenge [163], the problem was to predict the probability of a user engaging with Twitter interactions such as like, reply, retweet, and quote tweet. The winning solution used a variety of good features that represent the importance of users and message content [164]. However this challenge problem is different from our simulation problem for two main reasons. First, it assumes both a user and a message exist in the testing period. This is an impractical assumption to make when simulating a social system. Simulators do not assume to have any prior knowledge in the testing period. Second, it does not ask *when* such interaction is going to happen. For example, a classifier can predict the binary interaction between a user and a message but it assumes such interaction may happen sometime in the future without directly specifying it. In contrast, a reliable simulator should predict the timing of such interactions more accurately.

In summary, there are various techniques developed to improve recommendation systems in general, but not many evaluated on social networks. The challenge here is how to develop recommendation algorithms for users whose interests change over time [161].

4.1.4 Network Link Prediction

Predicting future links in a social network is a popular task in the social science community. Recent survey papers on link prediction [165] and social influence prediction [166] review a decade of research in this field. There are two types of link prediction tasks. They include predicting missing links in a static graph, and ranking most probable links in a future snapshot of a dynamic graph. Traditional models use hand-crafted features to achieve good performance. Distance-based features

(e.g., shortest path), and neighborhood features (e.g., the number of common neighbors, Jaccard's coefficient, Adamic-Adar index, Katz index) are the most effective features in the link prediction tasks [165].

Most recent network link prediction methods use Graph Neural Networks (GNN) [167]. The major benefit of GNN methods is the ability to make predictions for nodes unseen in training. GNNs achieve this capability via representing nodes with the features extracted from the local network neighborhood structure. This feature extraction process is done automatically by learning a function to aggregate the local network's neighborhoods information. Once learnt, this function is able to distinguish different node's local neighborhood structures. Many GNN methods (e.g., GCN [168], GAT [169], GraphSAGE [170]) are proposed in the literature. The key distinctions among many GNN methods are in how different approaches aggregate local network neighborhoods information.

Despite the recent success of GNN methods in the network link prediction literature, they have limitations. Theoretically it is proven that GNN methods are not significantly more powerful than Weisfeiler-Lehman (WL) graph kernels [171]. WL kernel is a simple iterative neighborhood aggregation method that was widely used to solve the graph isomorphism problem [172]. Another concern is due to the benchmark graph datasets that are typically used to evaluate GNN methods. For example, a recent study [173] showed that combining label propagation and feature augmentation techniques can beat the GNN performance on homophilous graph datasets that are widely used in the GNN literature.

In summary, there has been a significant advancement made in the network link prediction research both in terms of the features and the techniques (e.g., GNN). However, many traditional predictive models are built only for the nodes who are already seen in the training data (*transductive*

capability), but not for the new nodes who only appeared in the testing data (*inductive capability*). While recent deep learning methods overcome this issue to some extent [170], more experiments are required to test the robustness and generalizability of these methods over different types of social networks.

4.1.5 Simulating Finer Grained Social Media Activity

The studies related to predicting finer grained social media activity are lacking. Most recent attempts in this space are part of the Computational Simulation of Online Social Behavior (SocialSim) program sponsored by the Defense Advanced Research Projects Agency [174]. Abdelzaher et al. [107] proposed SocialCube, an agent-based approach to predict social media activities. This solution decides optimal agent-specific configurations from past social media traces. Garibay et al. [175] proposed DeepAgent to simulate the social media activity in the population, community, user, and content levels. This framework used a generative rule-driven approach where specific rule sets were built to model agent behavior using both endogenous and exogenous signals. While we have similar objectives, our solutions are not composed using specific individual agents' actions or hand-crafted rule sets.

4.2 Simulation Scenarios

The main goal of having a reliable social media simulator is to foresee the activities of a social media platform more accurately than what can be predicted by chance [176], or what can be judged by a human operator [177].

This dissertation focuses on two scenarios that motivate the design of the social simulators that we developed. We classify these scenarios by the endogenous and exogenous input features that

Table 4.1: The overview of the simulation scenarios.

Input Features	Output Granularity	Evaluated Context	Platform
Endogenous	*Who says to Whom When*	Organic discussions on cyber-security and crypto-currency community	Reddit
Endogenous and Exogenous	*Who says What to Whom When*	Political crisis in Venezuela	Twitter

we used to train a simulator (as shown in Table 4.1). We discuss the design and implementation of these simulators in the subsequent chapters.

Chapter 5: Simulating Online Discussion Threads Using Endogenous Signals[3]

Discussions on social media have significant impact on society. From recruitment to political movements to disinformation campaigns, social media discussions are the driving mechanism for information diffusion and user engagement. A particular variation of online discussions is a conversation tree, as seen on Reddit, StackOverflow, or Digg. In Reddit, for example, conversations are grouped on user-defined topics (often known as subreddits). The root of the conversation tree is an original post by a registered user; users respond with comments to the original post or other users' comments, repeatedly getting involved in the same conversation. Messages are often repeatedly exchanged between two users in a long conversation chain [135]. Discussions may lead to provocative, offensive, or menacing comments that end up involving an increased number of reactions and users [132].

Forecasting how conversations will evolve on such platforms is useful to many applications. For example, while it is difficult to know how many users follow a conversation over time without contributing to it, the number of users who contribute can help estimate the number of users exposed to the conversation. This information can be used to trigger the intervention of a subreddit administrator, for example, if the original posts are predicted to create unwanted engagement such as a coordinated disinformation campaign that is not likely to pass unnoticed. Accurately predicting the group of highly engaged users is important for developing intervention techniques to control information or manipulation spread and to accurately gauge the community opinion.

[3] A part of this chapter was previously published in [178]. Permission is included in Appendix A.

One challenge of addressing this problem is that real environments consist of simultaneous conversations on related topics [146]. For example, a user can engage multiple times in the same discussion thread; the same user can participate in multiple related conversation threads, thus affecting the overall audience size; simultaneous related conversation threads might compete for the attention of the same users, thus impeding or accelerating their involvement. Thus, one needs to take the groups of simultaneous conversations into account when developing a reliable simulator.

Much of the previous work has focused on predicting isolated properties of individual social media conversations such as size [179], temporal growth [126], and virality [115]. However, these efforts assume to know the initial growth of a conversation to predict the property of interest in the remainder of the conversation. The initial growth of a conversation in the first few hours has been shown to be most useful in predicting the future growth of the conversation [114].

This chapter proposes a method for forecasting the ensuing conversations with timing and authorship properties when given a set of topic-related original posts in a continuous interval of time on a platform. The contributions of this chapter are the following:

- We predict the properties of conversations in a finer granularity that include *whether*, *when*, and by *whom* a comment will be made in response to a post or another comment. This contribution is evaluated in terms of conversation structure (size and virality) and user engagement over time.

- We focus on predicting groups of conversations instead of individual conversations. We show that this focus is beneficial in accurately predicting the collective behavior of users who participate in multiple conversations.

- In contrast to most related work, our method only assumes to know the original post (or root of individual conversations), without initial reaction information. Previous studies used the comments that a post receives in the first few hours to predict the remainder of the conversation.

This chapter is organized as follows. Section 5.1 presents our solution in detail. We describe the dataset that we use to evaluate our solution in Section 5.2. Performance results are presented in Section 5.3. Section 5.4 summarizes the contributions presented in this chapter.

5.1 Predicting Pools of Conversations

Our objective is to predict the microscopic properties of a set of possibly inter-related, simultaneous conversations over time. The operational scenario we are considering is the following: given the initial postings described by content, author, and time on a given social platform (such as the four messages depicted on the horizontal time axis in Figure 5.1), generate the emerging discussion threads by specifying which message is in response to which message, and the author and time of each message. Each discussion thread generated will be represented as a conversation tree, where a child node is a message in response to its parent node in the tree; users can engage repeatedly within a conversation; the delay in responding to a previous message is unbounded; and a user may respond to his own message, typically with additions or clarifications. Table 5.1 presents the terminology used in this chapter.

Our solution is as follows. We probabilistically generate pools of independent conversation trees rooted in each input seed. We assign users and timing information to all nodes in every conversation tree. We thus end up with naive groupings of independent conversations, where user and

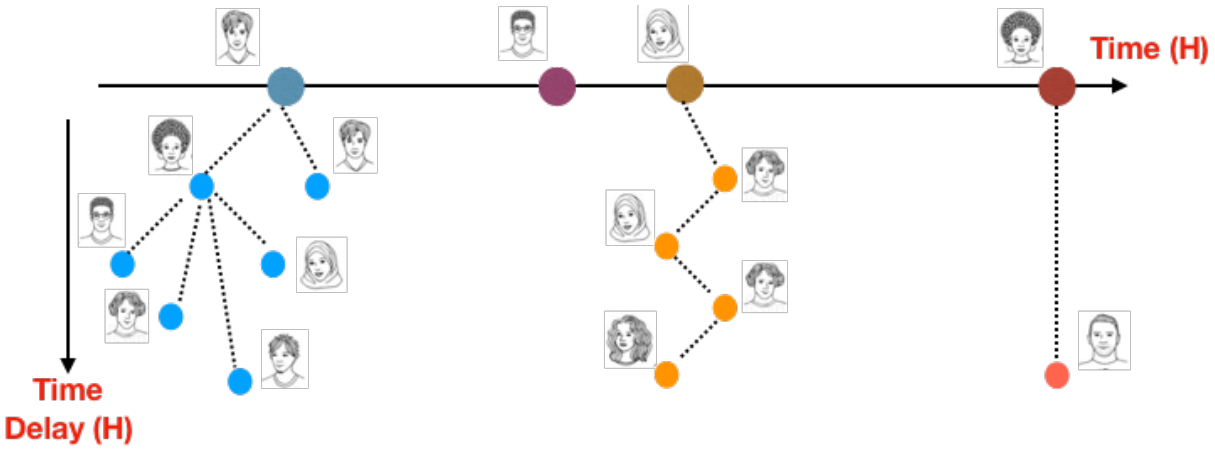


Figure 5.1: Sample simulation scenario. Given four original posts, the objective is to generate the corresponding conversation trees given that previously unseen users can engage in conversations; messages may be posted with unbounded delay; some original posts will remain unanswered; the conversation trees will have highly different structures; users may engage repeatedly with the same or different conversations.

Table 5.1: Terminology used in this chapter.

Term	Description
Node	Message in a discussion thread described by content, author, and posting time
Conversation tree	A conversation thread represented as a tree of messages, as shown in Figure 5.2a
Conversation pool	A collection of conversation trees within a finite period
Conversation size	The number of messages in a conversation
Conversation pool size	The total number of messages in all the conversations in a pool
Conversation depth	The number of levels in a conversation tree
Conversation breadth	The number of messages in a given level of a conversation tree
Node degree	The number of immediate messages in response to the parent message
Seed	A message at the root of the conversation tree
Propagation delay	The time difference between the posting of a message and that of its parent
Structural virality	Wiener Index of a conversation tree [180]
Collectivity	Group behavior of users engaged in multiple conversations [181]

time assignment to messages in a conversation are oblivious to what happens in other conversations in the same pool (Section 5.1.1). We then use a genetic algorithm to reconstruct a realistic pool of conversations from the arbitrarily generated ones (Section 5.1.2).

5.1.1 Generating Pools of Conversations

We employ the branching model [182] to construct pools of conversations. We are building on research [26] that shows that branching models based on node degree distributions can be used to accurately generate sub-trees of conversations. In this work, we extend this technique to generate temporal conversation structures of any depth while attaching user information.

We build each conversation tree recursively, as presented in Algorithm 1. The steps of this algorithm are as follows. From the training dataset that contains a large number of conversation trees, we build degree distributions per level. Thus, for each level, we will have a degree distribution for the nodes located at that level across all conversation trees. The node degree is defined as the number of children of that node in the conversation tree. Given an initial seed that functions as the root of the conversation tree to be generated, we recursively build tree structures by selecting node degrees from the degree distribution of the corresponding level. For a set of n input seeds, we thus generate n independent conversation trees that we consider a pool.

In order to assign authors to nodes in a conversation tree, we exploit the social network topology of previous user interactions. Specifically, from the training dataset, we extract the interaction network in which vertices are users and directed edges represent previous interactions. We also extract edge weights that represent the number of previous interactions. Note that a user can be part of her own neighborhood if she replied to her own post in the past. This is reflected by a

Algorithm 1 Probabilistic Generation of a Conversation Pool

PREREQUISITES: degree distributions per level of a conversation
INPUT: parent node
OUTPUT: a conversation tree

```
1: function GENERATE_CONVERSATION(parent, conversation)
2:     if parent is NULL then
3:         root_level ← 0
4:         root_degree ← SAMPLE_DEGREE(root_level)
5:         parent ← Node(root_level, root_degree)
6:         conversation.set_root(parent)
7:     level ← parent.get_level()
8:     N_children ← parent.get_degree()
9:     for j ← 1 to N_children do
10:        child_level ← level + 1
11:        child_degree ← SAMPLE_DEGREE(level)
12:        child ← Node(child_level, child_degree, parent)
13:        conversation.set_child(child)
14:        return GENERATE_CONVERSATION(child, conversation)
```

INPUT: $seeds_1 \ldots seeds_N$
OUTPUT: a pool of conversations

```
    function GENERATE_CONVERSATION_POOL(seeds[])
2:      seed_size ← length(seeds)
        conversation_pool ← []
4:      for k ← 1 to seed_size do
            conversation ← Tree()
6:          conversation.set_root(seeds_k)
            GENERATE_CONVERSATION(seeds_k, conversation)
8:          conversation_pool[k] ← conversation
        return conversation_pool
```

weighted self-loop in the network. We use this directed, weighted interaction network to bias the assignment of users to messages as follows. We start with a conversation tree, as generated above, whose root has a user assigned (from the input data). Recursively, for every node with a user u assigned, we probabilistically select d users from u's neighbors $N(u)$ in the interaction network and assign them as authors to the node's d children. If $d > N(u)$, we add $(d - N(u))$ new users who

were previously not seen in the training data to the chain of responses. We bias the probabilistic selection using the weights in the interaction network. Note that this approach allows for the same user to participate multiple times in the conversation tree.

In order to assign time to nodes in the conversation tree, we use a propagation delay distribution conditioned by the size of the conversation. We consider the propagation delay as the difference between the time of each comment and the time of parent comment/post in the training dataset. For each conversation, we extract the size of the conversation and the sequence of propagation delays. In the generated conversation, we use the size of the conversation resulting from the generation process (Algorithm 1) to randomly select a sequence of propagation delays from a previously seen conversation of that size. We sort the nodes of the generated conversation by level, assign the propagation delay to nodes, and compute the message time using the time of the seed message and the assigned propagation delay.

After this procedure, we end up with conversation trees rooted in the original message from the input data, in which each message node has a user and a time assigned. This simple probabilistic approach generates pools of independent conversations that ignore multiple aspects of real-world behavior, such as users participating in multiple conversations within the same period of time or, alternatively, being unable to participate simultaneously in many conversations. During empirical evaluations based on a variety of performance metrics that will be described later, we observed that all pools perform comparably and poorly when compared with testing data.

5.1.2 Reconstructing a Realistic Pool of Conversations

Ideally, given a set of possible pools of n conversations each corresponding to the n input seeds, we would construct a new pool consisting of the "best" conversation for each seed. However, there are two challenges. First, it is impossible to know which conversation is the best before the testing of the entire pool. This is mainly due to the huge variety of possible conversations that can be generated randomly.

Second, using a single performance metric that evaluates the "goodness" of individual conversations, selecting a pool of the best such individual conversations does not lead to a pool good enough in other metrics. For example, a pool constructed from the best individual conversations according to structural property metrics might evaluate poorly in user-level metrics.

To address these challenges, we treated the pool reconstruction problem as an optimization problem that we solved using a genetic algorithm. As the fitness function in the genetic algorithm, we used the output of two trained machine learning models to evaluate the goodness of a conversation.

5.1.2.1 Modeling the Problem using a Genetic Algorithm

Genetic algorithms provide powerful search heuristics for complex search spaces [183]. To proceed with the standard steps of genetic algorithms, we map our problem into the genetic algorithm context as follows: We consider a *gene* an individual conversation, represented by the message tree with assigned user and timing information attached to nodes. An *individual* in the genetic algorithmic representation is thus a pool of conversations in our context. The *population* is the set of conversation pools we generated with the probabilistic approaches described earlier. The objective is to create a pool of conversations that outperforms any existing pool of conversations.

We use the standard framework of a genetic algorithm and repeat the process until there is no improvement in the best solution. We start with the initial set of n conversation pools as described earlier. We measure the fitness of a conversation pool using two trained machine-learning models as described next. We rank the conversation pools according to the fitness function and consider only the top 80% for mate selection. Given a pair of conversation pools selected from a top-ranked pool and a least-ranked pool in this top 80% pools, we randomly draw conversations (without replacement) to form a new pool for the next generation. Thus, the new generation entirely consists of conversations from the top 80% of the conversation pools in the previous generation. Accordingly, we re-construct n new pools for each generation. We summarize all algorithmic steps in Algorithm 2.

We do not use mutations in this approach for the following reason. Mutations require modifying the initial conversation structures (with user and timing information) generated by the probabilistic model. The mapping of users to the internal conversation nodes is done via a recursive chain of user assignments using the interaction network. When we modify the structure, this method of mapping users becomes obsolete and leads to an inaccurate view of user responses.

5.1.2.2 Ranking Pools of Conversations with Machine Learning

In order to rank the pools of conversations, we assign a goodness score to each conversation in the pool and consider the sum of all such scores as the goodness score of the pool. The goodness score of each conversation has two components: a score relative to the structural properties (i.e., the shape of the conversation tree), and a score relative to the timing of the nodes in the conversation. Specifically, we feed each conversation into two trained machine-learning models to assess the goodness of the branching factor and propagation delay with respect to the attached user information and semantic structure.

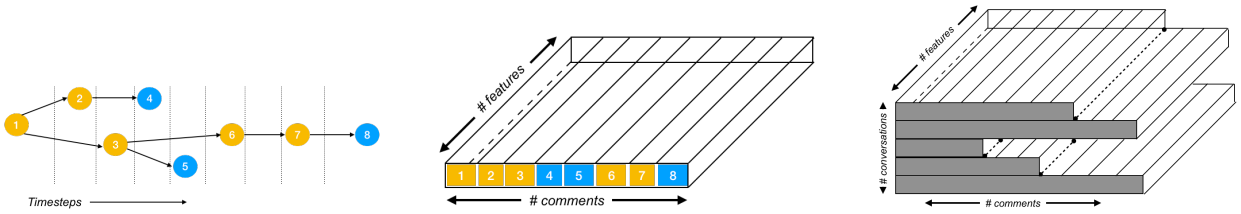
Algorithm 2 Selection of the Best Conversation Pool with a Genetic Algorithm

INPUT: a set of conversation pools, γ the probability of mate selection
OUTPUT: a set of re-constructed conversation pools

1: **function** NEXTGENERATION(P,γ)
2: $P \leftarrow$ RANK_POOLS(P)
3: $P_{mates} \leftarrow$ SELECT_BEST_POOLS(P, γ)
4: $P_{gen} \leftarrow$ RECONSTRUCT_POOLS(P_{mates})
5: **return** P_{gen}

INPUT: initial set of conversation pools
OUTPUT: best conversation pool

function GENERATE(P,γ, N_{Gens})
2: **for** $N_1 \leftarrow 1$ to N_{Gens} **do**
$P \leftarrow$ NEXTGENERATION(P, γ)
4: $P \leftarrow$ RANK_POOLS(P)
$pool \leftarrow$ SELECT_BEST_POOL(P)
6: **return** *pool*



(a) Conversation Tree (b) Individual Conversation (c) A Pool of Conversations

Figure 5.2: Representation of conversation trees. a) Nodes (messages) are ordered chronologically. Yellow nodes represent internal nodes and blue nodes are leaf nodes. b) Each node is represented by a spatio-temporal feature vector. Feature vectors are ordered chronologically and grouped by conversation. c) Multiple conversations of arbitrary size are stacked together for training/testing.

We use two individual-level properties—branching factor and propagation delay—of conversation nodes as the target units for the prediction tasks. Any information regarding branches is important for the accurate creation of the conversation structure as they evolve in the form of sub-trees under the same original post or another comment [132]. Therefore, we first classify the messages as leaf or branch nodes in the tree. Note that these node positions determine the shape of the conversation.

We classify the messages by the delay with which they are posted in response to their parents to distinguish fast-paced conversations from slow-paced conversations. We consider the median propagation delay within a conversation as the borderline between the two classes: messages with a propagation delay larger than this median are called late adopters, while the others are early adopters. We used this binary classification approach to seek the hourly time granularity predictions. We discovered empirically that the median propagation delay is close to 1.5 hours and a binary classification satisfies the hourly granularity. For finer time granularity, we might need to classify propagation delays by quartiles, or predicting the exact propagation delay value in seconds. This would remain as future work to improve the time predictions.

Each message is described by features from three main categories: i) spatio-temporal features, that capture the position of an individual message in a conversation, ii) user features, and iii) content features. These features are detailed in Table 5.4. We use the LSTM model to capture the chronological order of messages in a conversation. The input to the LSTM algorithm is a conversation as shown in Figure 5.2b. We use the memory-cell design of a standard LSTM in our work [184] which is implemented in Keras [185]. Our LSTM setup includes two blocks of memory-cells with 32 and 8 hidden units, and we use the Adam algorithm for the optimization with a learning rate of 0.001 based on hyper-parameter optimization. Conversation representations are different in shape mainly by the number of messages in the online conversation, and thus we input them one by one for training.

During testing, we extract the features described in Table 5.4 from the generated conversations. The activity-level features of the users in a particular conversation are constructed considering their activities in other conversations. To account for the interaction among multiple conversation

trees, we dynamically update the user features. Specifically, when a user j is assigned to a message in a new conversation tree at time t, her activity features such as the number of past activities $A_j^{t'}$ at time $t' < t$ is updated to $A_j^t = A_j^{t'} + 1$. Since we do not predict the content of the messages in a conversation, we assign content-level features to messages in the testing period randomly based on previously seen conversation nodes in the same level.

Once we construct the data structure shown in Figure 5.2b with all necessary features, we infer one binary vector that represents branch/leaf using the branch discriminator model, and another binary vector that represents the early/late adopters using the delay discriminator model. We consider these two binary vectors as the inferred ground truth to assess the generated conversation. The assessment is done by comparing the inferred ground truth with the same binary vectors extracted from the generated conversations using the area under the curve (AUC). Each conversation receives a goodness score as the mean of two AUC scores from the two models. The goodness of a pool of conversations is the sum of the goodness scores of the conversations in the pool. We use this pool goodness score to rank the pools of conversations in the genetic algorithm (as shown in RANK_POOL function in Algorithm 2).

5.2 Datasets

For empirical evaluations, we focus on Reddit conversations. We selected two active topics, crypto-currency, and cyber-security, as our two topic-driven separate datasets. We extracted all conversations between January 2015 and August 2017 posted under the topic-related subreddits and listed in Table 5.2. Both datasets were provided privately as part of the DARPA SocialSim program.

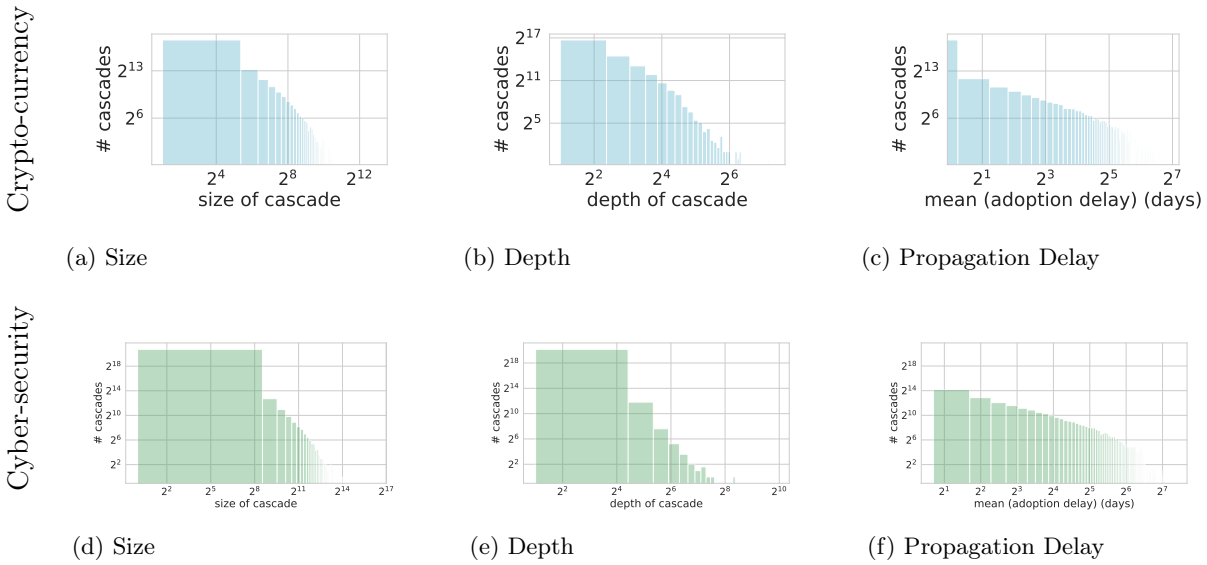
Table 5.2: Subreddits used for data collection. We collected 0.2M conversations from 9 subreddits related to crypto-currency and 1.76M conversations from 38 subreddits related to cyber-security.

Domain	List of Subreddits
Crypto-currency	/r/Bitcoin, /r/Ethereum, /r/Monero, /r/Paycon, /r/DopeCoin, /r/Lisk, /r/Donationcoin, /r/Pivx, /r/Orocoin
Cyber-security	/r/netsec, /r/netseclounge, /r/technology, /r/techsupport, /r/pcmasterrace, /r/linux, /r/hacking, /r/Piracy, /r/sysadmin, /r/HowToHack, /r/privacy, /r/Windows10, /r/programming, /r/networking, /r/softwaregore, /r/compsci,/r/talesfromtechsupport, /r/msp, /r/security, /r/SocialEngineering, /r/Malware, /r/AskNetsec, /r/blackhat, /r/ReverseEngineering, /r/crypto, /r/pwned, /r/netsecstudents, /r/securityCTF, /r/hacktivism, /r/browsers,/r/linuxadmin, /r/websec, /r/antivirus, /r/Ransomware, /r/Pentesting, /r/OpenHacker, /r/blackhatting, /r/Android

We represented each conversation thread as a conversation tree. A node in the conversation tree consists of the textual content of a Reddit message (post or comment) and its author. A pair of nodes (source to target) are connected by a directed edge where the direction suggests that the target node reacts to (content posted by) the source node. Table 5.3 presents the structural properties of the conversations in the two datasets. The cyber-security dataset is nearly 10 times the size of the crypto-currency dataset in the total number of messages posted. The properties of the conversation trees are also highly different in scale: the largest conversation in cyber-security contains 74K messages, while in crypto-currency is 7.8K. The depths of the conversation trees are different: 971 vs. 160. Irrespective of the size and depth disparities, we observe that Reddit conversations are viral and broad. They include both slow (cyber-security) and fast-paced (crypto-currency) conversations which can be active for short and long periods. (as seen in Figure 5.3). Moreover, the discussions that originate from crypto-currency subreddits exhibit diverse characteristics related to the scale and speed of discussion spread [186].

Table 5.3: Properties of Reddit conversations in our datasets.

Measurement	Crypto	Cyber
Number of conversations	209,721	1,762,977
Number of messages	3,580,162	35,381,971
Number of distinct users	144,457	1,647,789
Max conversation lifetime (days)	311	910
Max conversation size	7,868	74,032
Max conversation depth	160	971
Max conversation breadth by level	7,578	72,955



(a) Size (b) Depth (c) Propagation Delay

(d) Size (e) Depth (f) Propagation Delay

Figure 5.3: Basic characteristics of Reddit conversations. The distribution of cascades is presented by (a,d) size, (b,e) depth, (c,f) the mean delay between the time of a comment and the time of the original post as observed in the conversations.

Figure 5.4 depicts a sample group of conversations on Reddit related to the Bitcoin scaling debate [187] from August 2017. The debate eventually led to the creation of a new crypto-currency, *Bitcoin Cash (BCH)*, along with a new software repository on GitHub that implemented the scale-up solution. There are users who repeatedly participate in the same and multiple conversations during the debate. For example, there are 57 conversations with 4,418 messages posted by 1,458 users. 218 and 83 users appeared in more than one, and two conversations, respectively.

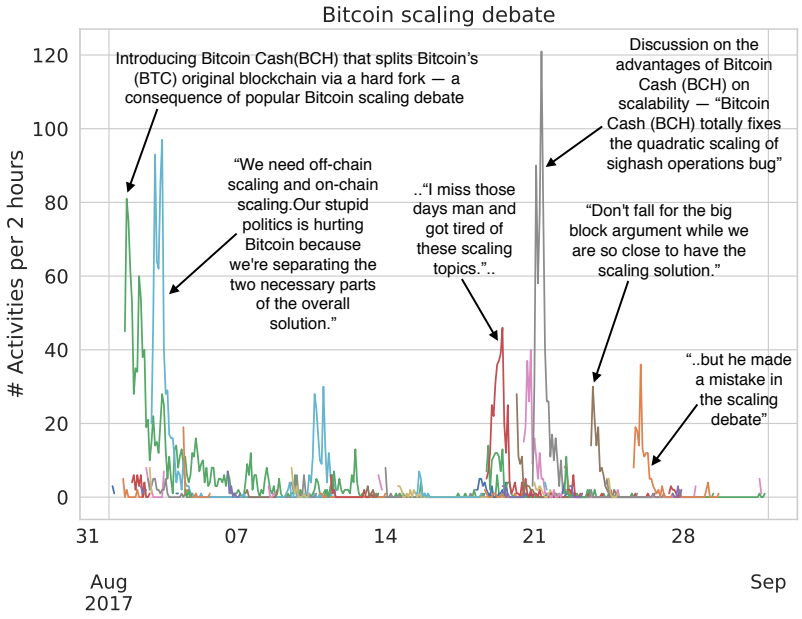


Figure 5.4: Discussions on Reddit during the Bitcoin scaling debate.

From these datasets we extract three groups of features (detailed in Table 5.4): i) spatio-temporal features, ii) user-level features, and iii) content-level features. We represent the topology around an individual node in the conversation using two spatio-temporal properties: degree and the birth order of the predecessors. As an example, we use the degree and birth order of the parent (level $i-1$) and the grand-parent (level $i-2$) nodes to represent a node in level i.

Actions in a conversation could be in response to the users who authored the previous message rather than simply to the content with which the users interact. We thus represent a user via a set of features describing her status on the platform, measured by the amount of activity she has done before the particular reaction. Such activities reflect the user's interest in other conversation threads. We also extract the popularity of the user in terms of *upvotes* and *downvotes* received to her posts or comments in the past. These endorsements summarize the influence of a user in a community.

Table 5.4: Features used to represent a message in a Reddit conversation.

Feature Domain	Feature Description
Structural Features	Number of comments for comment/post
	Adoption delay from the parent comment/post
	Adoption delay from the root post/root
	Level of the conversation tree
	Birth order of comment
	Number of comments for the parent comment/post
	Birth order of the parent comment
	Number of comments to the grandparent comment/post
	Birth order of the grandparent comment
User Features	Total number comments received by the comment author in the past
	Total netscore (upvotes−downvotes) of the comment author in the past
	Total number comments made by comment author in the past
Content Features	Netscore of the comment
	Subjectivity score of the comment
	Controversiality score of the comment
	Netscore of the parent comment
	Subjectivity score of the parent comment/ post
	Controversiality score of the parent comment
	Netscore of the grand parent comment
	Subjectivity score of the grand parent comment/ post
	Controversiality score of the grand parent comment

We extracted the sentiment scores of Reddit comments that quantify the subjective and controversial content (a Python library of a natural language toolkit is used to calculate these two features [188]). We also captured the semantic structure of the comments at predecessor nodes. Another useful feature is the popularity of posts or comments that is captured by *net-score*, the difference between up-votes and down-votes received for a particular post or comment from all users.

5.3 Evaluation

The primary objective of the generative model proposed in this study is to predict the complete conversation structure with authors and timing information. For a comprehensive evaluation, we compare the following outcomes against the ground truth conversations: (i) the structural char-

Table 5.5: Reddit conversations grouped by post time.

Domain	Training (Jan '15–Jul '17)		Testing (Aug '17)	
	# Conversations	# Messages	# Conversations	# Messages
Crypto	0.19M	3.3M	0.02M	0.25M
Cyber	1.7M	34M	0.06M	0.9M

acteristics in terms of size and virality of the predicted conversations; (ii) the volume as measured in the number of comments generated to the seed posts and audience size as measured in the number of distinct users who participate in the conversations over time, and (iii) the collective behavior of users who engage in multiple conversations.

For testing the generated pools of conversations, we used a subset of the testing data as follows. We used as seeds the posts made between August 1 and August 3, 2017, and the resulting conversations as seen by the end of August 2017. There were 3,740 and, respectively, 3,463 such conversations in the crypto-currency and cyber-security domains. Because seeds are chosen from a continuous time interval, the ensuing conversations can overlap in time.

We compare the quality of our model with respect to three baseline models. First, we use a state-of-the-art generative model (i.e., Lumbreras Model [131, 189]) that predicts the entire structure of the conversation instead of aggregate metrics such as size or virality. The *Lumbreras model* proposed an improved solution compared with a family of generative approaches [190, 191] that use the branching process in the generation of conversation structures. A more recent work [135] that adds reciprocity as a model parameter acknowledges increased computational costs relative to previous work due to various optimization functions. Due to the size of our datasets, we chose to compare with the less computationally intensive Lumbreras model. This model uses parameters related to popularity, novelty (preference to reply to a newer post), root-bias (preference to reply to a post rather than to a reply itself), and user roles to predict the growth of discussions. We

construct 10 pools of conversations from this solution to account for the bias in parameter selection criteria. However, this model does not assign user information and maps only discrete timestamps to the generated comments. We do not use the Lumbreras model in the temporal measurements due to the mismatch between our continuous time and its discrete time approaches.

Next, we use two baseline models that draw the events from the training data repeatedly into the testing time period. *Baseline (recent-replay)* draws the most recent n conversations from the training data. *Baseline (random)* draws n conversations from the training data at random (where n is the number of seeds in the testing period). We construct 10 pools of conversations in the Baseline (random) solution to minimize the bias of random selection. In the baseline solutions, we keep all other event information (e.g., author, the conversation structure, etc.) of the conversations except the event timestamps, which are shifted by the time interval between the seed post and their corresponding root message. Because these baseline models repeat events from the recent past, they proved to be very challenging to outperform in simulating user activities in multiple social platforms [107, 106], including Reddit [128].

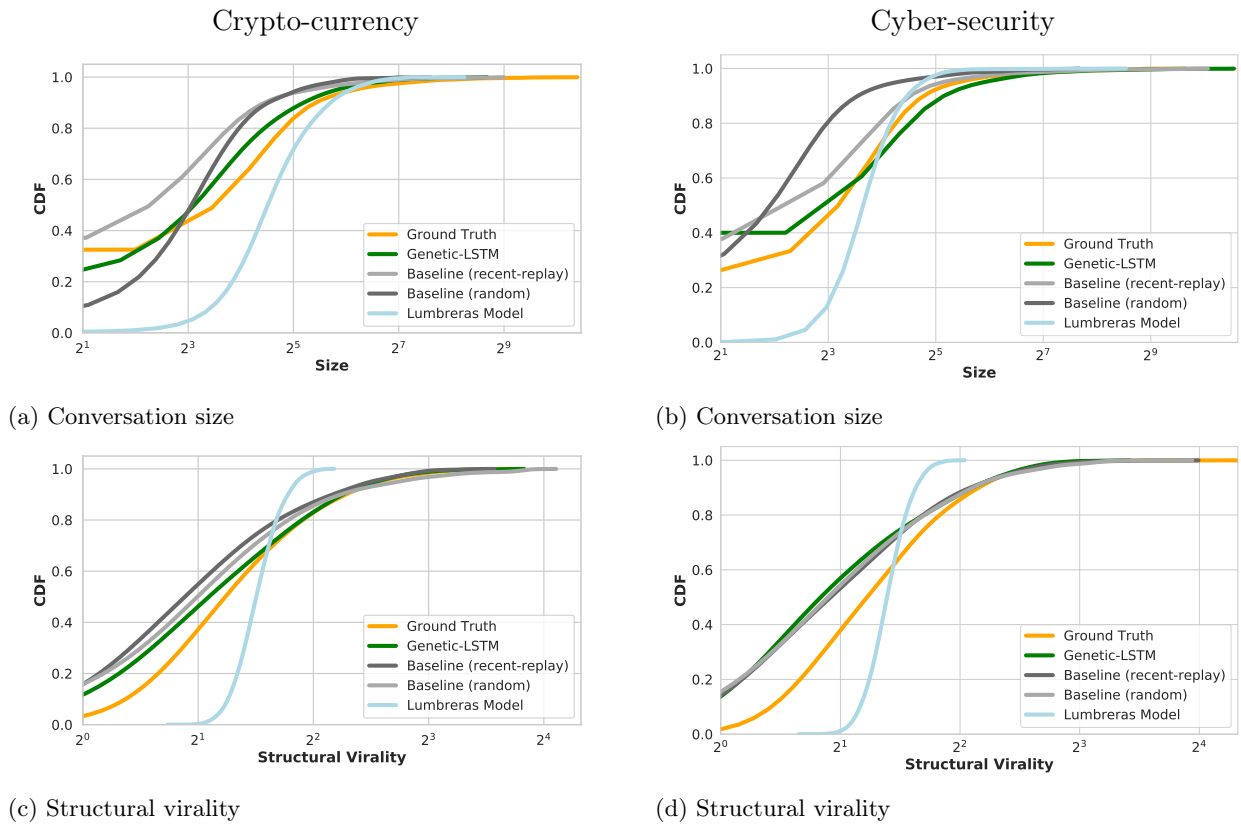
To evaluate the accuracy of our conversation reconstruction solution, we use several measurements. First, we evaluate the goodness of our fitness score used in the conversation reconstruction algorithm (Section 5.3.1). Second, we present the structure of conversations in the reconstructed pool with respect to size and virality (Section 5.3.2). Third, we evaluate the volume of messages generated from the original posts with respect to the community of users who authored them and timing information (Section 5.3.3). Finally, we quantify the engagement of users in multiple conversations (Section 5.3.4). These metrics are reported in comparison with ground-truth data and the baseline models mentioned above.

5.3.1 The Goodness Score of a Conversation

We measure the two components of the goodness score: predicting the position of a message as a branch or leaf node in the conversation tree, and the timing of the message as early or late compared to the median propagation delay relative to that conversation. We train four LSTM models in total for two training datasets as described in Table 5.5. The outputs of these LSTM models are used to assess the likelihood of a conversation in the conversation reconstruction algorithm. LSTM-degree models achieve a 73-75% F1 score in discriminating leaves vs. branching nodes in respective domains. A majority vote would achieve 65% accuracy on predicting branches as the two classes are balanced in the ratio of 65%:35% across both datasets. The F1 score of our LSTM-delay models in distinguishing between early and late adopters is 83-89% while a random draw should achieve 50% given the perfectly balanced classes.

5.3.2 The Structure of Conversations in the Pool

To measure the size and structural virality of the generated conversations irrespective of the temporal aspects, we compare the re-constructed conversation pool with the baseline generative approaches. We show the Cumulative Distribution Functions (CDF) of individual conversation sizes and structural virality scores for conversations resulting from our model, the baseline approaches, and the ground truth in Figure 5.5. For fairness in evaluating the baseline approach, for the Lumbreras model and Baseline (random) we generated 10 solutions for each seed and reported the average. We calculate the absolute percentage error (APE) of the mean size and the mean structural virality between the generated conversations and the ground truth conversations. We also report the JS divergence between the distributions of the structural metrics reported from the generative



(a) Conversation size

(b) Conversation size

(c) Structural virality

(d) Structural virality

Figure 5.5: The distribution of the size and virality of conversations.

models and the ground truth (as shown in Table 5.6). A lower JS divergence value denotes that the distribution of the sizes/structural virality of the generated conversations is closer to that of the ground truth. We have three observations from these measurements.

First, our solution achieves the lowest JS divergence value after comparing the distributions of sizes and virality scores between the predicted and the ground truth conversations (as shown in Table 5.6). We also record the mean conversation size closer to the ground truth value across both datasets as shown by the lowest APE values for sizes in Table 5.6.

Second, we noticed that the mean structural virality scores of the conversations generated by our solution are closer to the ground truth in crypto-currency related discussions (lowest APE

Table 5.6: Performance of the size and structural virality of the conversations. We compare the distribution of size and virality of the generated conversations with the ground truth using JS Divergence (JSD). We also report the APE for the mean size and structural virality of the conversations after compared with the respective values in the ground truth. We highlight the lowest JSD and APE values in bold.

Domain	Model	Size		Virality	
		JSD	APE	JSD	APE
Crypto	Baseline (recent-replay)	0.40	51.7	0.043	17.6
	Baseline (random)	0.14	43.5	0.074	23.7
	Lumbreras Model	0.49	37.4	0.046	11.8
	Genetic-LSTM (our solution)	**0.15**	**25.4**	**0.012**	**7.5**
Cyber	Baseline (recent-replay)	0.39	28.9	0.035	14
	Baseline (random)	0.41	57.6	0.036	62.7
	Lumbreras Model	0.34	12	0.062	**0.3**
	Genetic-LSTM (our solution)	**0.23**	**8.6**	**0.029**	15.7

values for virality in Table 5.6) more than the cyber-security related conversations. We believe this is due to the slight over-prediction (12%-18%) of the number of smaller conversations (i.e., conversations with a size smaller than the median size) compared to what exists in the ground truth. The majority of smaller conversations only have immediate comments to the original post, thus the virality scores are very low.

And finally, we also notice the difficulty of accurately predicting the properties of the largest and most viral conversations. Note that the most viral conversation may not be the largest conversation [180]. For example, the size of the most viral (virality = 12) conversation is 136, and the virality of the largest conversation (size = 1,301) is 5 in crypto-currency discussions. We do not accurately predict the size and virality of such conversations compared to other baseline models (as shown in Table 5.7). However, we noticed those baseline models are not consistent in achieving the best results across crypto and cyber discussions. These conversations are very rare to observe and are likely to grow under external events [180, 192]. These external events may be in the form of crypto-currency prices, cyber-security attacks, or news events as reported by journalists. Our prob-

Table 5.7: Performance of the largest and the most viral conversations. We highlight the lowest APE values in bold.

Domain	Model	Largest	Most viral
		Size (APE)	Virality (APE)
Crypto	Baseline (recent-replay)	62	17
	Baseline (random)	**10**	83
	Lumbreras Model	113	**0**
	Genetic-LSTM (our solution)	69	8
Cyber	Baseline (recent-replay)	**34**	**21**
	Baseline (random)	121	147
	Lumbreras Model	358	53
	Genetic-LSTM (our solution)	87	47

abilistic model does not account for these external events on generating the conversation structure, thus it is unable to reproduce the properties of the most viral conversation. We plan to incorporate external events for modeling conversations in future work.

In conclusion, while our solution more accurately traces both the distribution of conversation sizes and that of conversation viralities than any of the baselines, it struggles with the endpoints of the spectrum: very small and very large conversation properties. However, we can conclude that we generate a pool of conversations that are closer to forecasted activity than simply representing the past through random sampling because in all metrics we consistently outperform the random and recent-replay baselines. The challenges posed by the two baseline models extracted from training data are evident also in comparison with the performance of the Lumbreras model: only once does the Lumbreras model outperform both baselines in terms of JS distances (Table 5.6). In terms of APE values (as presented in Table 5.6), it competes closely with the baselines.

5.3.3 Temporal Conversations

We compare the reconstructed pool of conversations with the ground truth data on different temporal measurements. We compare i) the size of the conversation pool as measured in the overall number of comments generated to the seed posts, and ii) the number of distinct users who participate in the conversation pool over time. We report Dynamic Time Warping (DTW) and Root Mean Square Error (RMSE) on these measurements between the conversations in the reconstructed pool and the conversations in the ground truth. We use daily granularity to bin the timeseries for comparison, and group these timeseries into five time intervals of 1–5 days, 5–7 days, 7–14 days, 14–21 days, and 21–28 days for a deeper evaluation.

Table 5.8 shows the APE values for the number of messages and the number of distinct users after comparing different models with the ground truth. Our simulations result in better estimations of the total number of messages than any of the baselines, with 25.3 and 8.5 absolute percentage error (APE) in the two datasets, which leads to 35%-50% performance gain over the best-performing baseline. However, our solution does not achieve the lowest APE on the total number of distinct users as we over-predict the number of users who participate in these conversations.

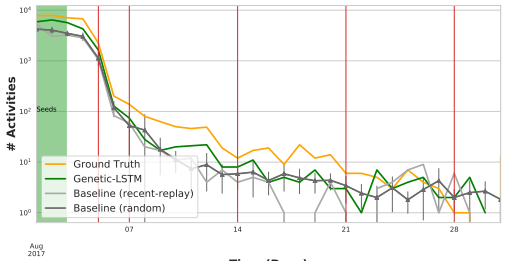
Table 5.8: Performance of the volume and users in the conversation pool. We do not report the number of distinct users for the Lumbreras Model as it does not predict user assignments. We highlight the lowest APE values in bold.

	Model	# Messages (APE)	# Users (APE)
Crypto	Baseline (recent-replay)	52	29
	Baseline (random)	50	**22**
	Lumbreras Model	37	-
	Genetic-LSTM (our solution)	**25**	36
Cyber	Baseline (recent-replay)	29	**2**
	Baseline (random)	58	27
	Lumbreras Model	11	-
	Genetic-LSTM (our solution)	**8**	67

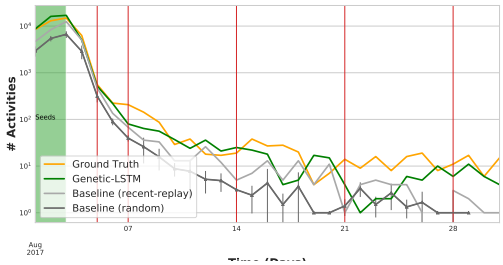
We are interested, however, in evaluating our predictions over the simulated time. This is particularly relevant for application scenarios such as designing intervention techniques when one would like to investigate "what if" scenarios and their consequences at particular times. Figures 5.6, 5.7, and 5.8 report the timeseries and the performance of predicting the volume of comments and the number of distinct users who participate in these conversations. There are multiple observations to be made from these plots. First, the trend of the number of messages and distinct users over time holds for our simulations and the baselines. This is because all models capture the intuitive phenomenon of high activity and user involvement when a post is freshly made, and the decay in interest as time passes.

Crypto-currency

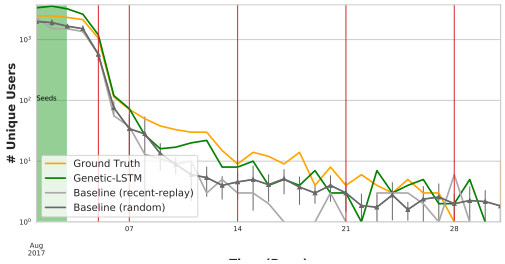
Cyber-security



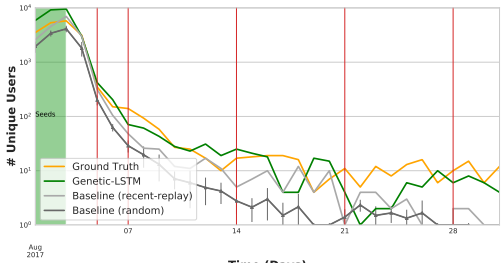
(a) Conversation pool size over time



(b) Conversation pool size over time



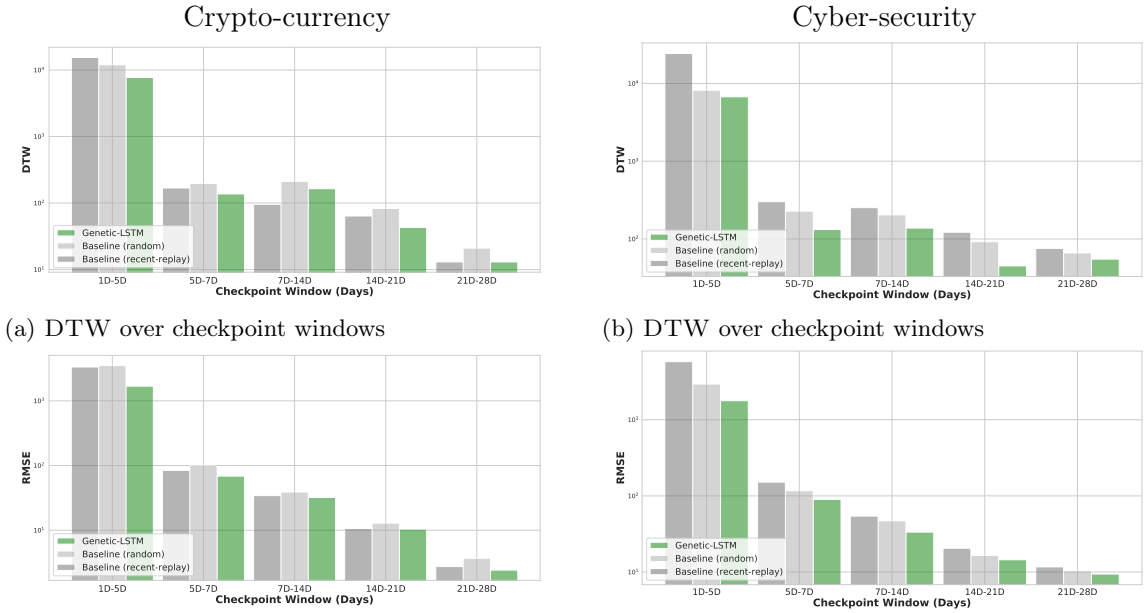
(c) Number of unique users over time



(d) Number of unique users over time

Figure 5.6: The conversation pool over time. We report the size of the conversation pool and the number of unique users participating in conversations over time for crypto-currency and cyber-security discussions. Genetic-LSTM (our solution) is compared with two competing baseline models, Baseline (recent-replay) and Baseline (random). Baseline (random) predictions are normalized over 10 different runs, and the error bars are reported for the standard deviation.

Second, our solution fares better than the baselines not only in the aggregate number of messages at the end of the simulation period but also over time: the green lines in Figures 5.6a and 5.6b are generally the closest to the ground truth plots in yellow. As shown in Figure 5.7c, our solution records a RMSE value of 1,685 compared to the RMSE values of 3,697 and 3,329 for the two baseline models on predicting the conversation pool size during the first five days (1D-5D). During the next time intervals, our solution records 2%-39% performance benefit in RMSE values over both datasets compared to the best-performed baseline solution (as shown in Figures 5.7c and 5.7d).



(a) DTW over checkpoint windows

(b) DTW over checkpoint windows

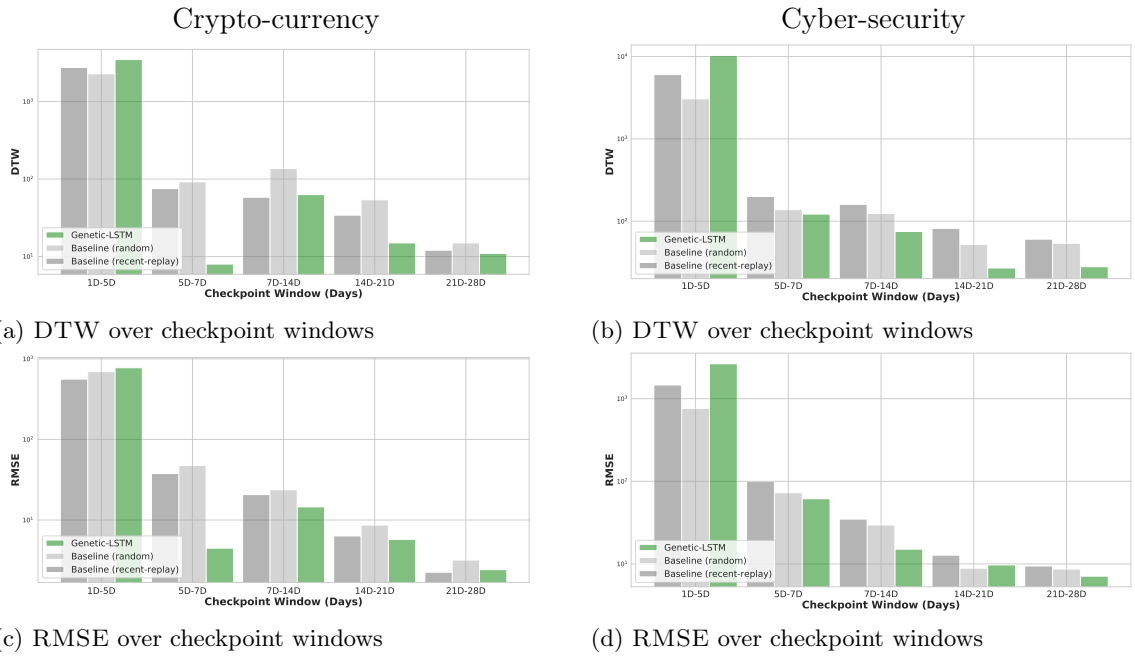
(c) RMSE over checkpoint windows

(d) RMSE over checkpoint windows

Figure 5.7: The size of conversation pool over time. We report the performance of predicting the size of the conversation pool over time for crypto-currency and cyber-security discussions using two quantitative metrics, (a,b) Dynamic Time Warping (DTW) (lower is better), (c,d) RMSE values (lower is better) after comparing each model predictions with the ground truth over different time intervals. Checkpoint windows are in days (D).

Third, our performance advantage over the baselines is higher in the cyber-security conversations, where our solution is always better than both baselines in both RMSE and DTW measurements for all interval periods shown in Figures 5.7b, and 5.7d. This is probably due to the significantly

larger dataset in cyber-security which is 10x larger than the crypto-currency dataset. A larger dataset generally helps our machine learning models to train and make better predictions. In general, our improved performance over baselines is likely due to incorporating original post information in generating the conversations and optimizing branching factor and propagation delay in the predicted pool of conversations. The baseline models do not account for such attributes but only replay the past events.



(a) DTW over checkpoint windows

(b) DTW over checkpoint windows

(c) RMSE over checkpoint windows

(d) RMSE over checkpoint windows

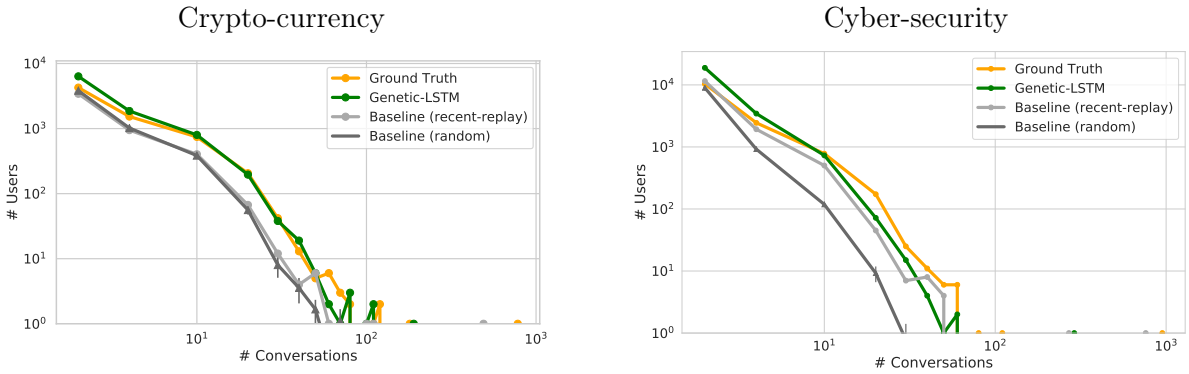
Figure 5.8: The number of unique users over time. We report the performance of predicting the number of unique users over time for crypto-currency and cyber-security discussions using two quantitative metrics, (a,b) Dynamic Time Warping (DTW) (lower is better), (c,d) RMSE values (lower is better) after comparing each model predictions with the ground truth over different time intervals. Checkpoint windows are in days (D).

And finally, our model performs better than the baselines in the number of users engaged over time in these conversations. For Reddit-like conversations, this is a challenge since discussions may lead to provocative, offensive, or menacing comments that end up repeatedly involving a subgroup of users [132]. For example, there are only 6,818 users who participate in 32,533 comments in

crypto-currency discussions. In the largest conversation, the ratio between the number of comments and the number of users is 2.35 in the ground truth, and 2.06 in our solution. Our model tends to over-predict the number of users engaged a short time after the seed messages are posted (as shown in Figures 5.8c and 5.8d for the interval 1D–5D), and consistently performs well for the more distant future. As shown in Figures 5.8a - 5.8d, our solution achieves the lowest DTW and RMSE values for the interval 5D–7D across two datasets, respectively. This is particularly relevant because it shows our model's predictive power for longer-term simulations: from the 6th to the 28th day of the simulation period, our model consistently predicts better the number of users and the timing of their comments.

5.3.4 Collective Behavior

Another important characteristic related to user engagement is the co-engagement with various topics. Specifically, empirical studies [193] have shown coordinated campaigns run as troll farms or cyborgs, where groups of users engage in multiple related conversations to shift the opinion of the general audience.



(a) User Engagement (b) User Engagement

Figure 5.9: The number of users who engaged with conversations. We show the comparison with the baseline models. The values in the y-axis are binned by the intervals of 10 in the x-axis.

We report two measurements to capture the collective behavior of users who participate in these conversations. First, we present the number of users engaged in multiple conversations (as shown in Figures 5.9a and 5.9b). Specifically, we record the number of conversations that a user engaged with, and count the users who engaged with X number of conversations ($1 \leq X \leq 3740$). We noticed a heavy-tailed distribution, where few users engage in many conversations. We calculate the JS divergence between each models' distribution and the ground truth distribution. Lower JS divergence values reflect predictions closer to the number of actively engaged users observed in the ground truth.

Our solution achieves the lowest JSD value of 0.05 (crypto) and 0.07 (cyber) after comparing with the respective baseline models. We also predict the number of highly active users closer to the ground truth value than any other baseline solution. In the crypto-currency discussions, we predict 1,916 users who engage with more than two conversations, while there are 2,438 such users in the ground truth and 1,310 such users in the best-performing baseline solution. Our relative success is due to implicitly accounting for simultaneous conversations with possibly common users in our modeling of the problem as a pool of conversations. Specifically, our LSTM-based model that helps to select the best pool of conversations accounts for user participation in multiple conversations, thus is able to predict better the number of highly engaged users than a model that simply repeats the past.

Second, we evaluate whether users participate in these conversations as a group according to a metric (*collectivity*) proposed by [181]. We record user participation in conversations in a vector $[c_1, c_2, ..., c_n]$, where c_i indicates a binary value to reflect the user involvement in the i^{th} conversation. For this metric, we only consider the most active users who participate in at

least three conversations (on average, a user participates in two conversations in the ground truth dataset). The original paper [181] used the Pearson correlation coefficient to compare all pairs of binary vectors. The higher the correlation coefficient values, two users participate in the same set of conversations. They also used the Jaccard coefficient to compare the overlap of conversations between two users. According to their experiments, the Pearson correlation coefficient and Jaccard coefficient values are correlated. While we do not experiment with any other similarity metric (e.g., Hamming distance), we believe they would result in distributions similar to what was observed using the Pearson correlation coefficient or the Jaccard coefficient. In this work, we use the Pearson correlation coefficient to quantify the collective behavior of user involvements.

We calculate the JS-divergence and RMSE between the coefficient distributions of the simulation and the ground truth data (as shown in Table 5.9). Lower JS-divergence values reflect collective behavior closer to that measured from the ground truth. We achieve the lowest 0.07 and 0.12 JS-divergence values, and lowest 1,815 and 976 RMSE values for the respective domains after compared with the respective baseline models.

Table 5.9: A comparison of the collectivity scores. We report the collectivity scores of users who participate in multiple conversations. We show JS-divergence (JSD) and RMSE values after comparing each models' distributions of collectivity scores with the ground truth values. We do not report the number of these measurements for the Lumbreras Model as it does not predict user assignments. We highlight the lowest JSD and RMSE values in bold.

	Model	Collectivity	
		JSD	RMSE
Crypto	Baseline (recent-replay)	0.09	8036
	Baseline (random)	0.14	8210
	Lumbreras Model	-	-
	Genetic-LSTM (our solution)	**0.07**	**1815**
Cyber	Baseline (recent-replay)	**0.12**	1779
	Baseline (random)	0.23	3049
	Lumbreras Model	-	-
	Genetic-LSTM (our solution)	**0.12**	**976**

In summary, our experimental results show that in addition to accurately predicting the structural properties of individual conversations, predicting pools of conversations also leads to more accurate predictions of user involvement over time.

5.4 Summary and Discussion

This chapter introduces a generative technique for predicting a group of simultaneous conversations in social media. Our solution uses a probabilistic generative model with the support of a genetic algorithm and LSTM neural networks. We tested our technique on two topic-based collections of Reddit conversation trees. Given a set of posts in a continuous time interval, our solution generates the full set of reactions to each message, including reactions to reactions, without having access to, for example, intermediate states of the conversation tree. In addition to generating the structure of conversation trees, our solution also assigns authorship and timing information to each message. The code for this framework is available publicly [194].

Our solution captures the relationship between different microscopic conversation properties including the structure, propagation speed (timing), and the users who participate in a set of simultaneous conversations. We trained two LSTM models *on pools of conversations* to capture this relationship. In the first model, we predict whether a node in the conversation is branching (thus, generating more reactions) or is a leaf in the conversation tree. The second model classifies messages by the delay which they are posted in response to their parent. Both models use structural, user, and content features in the temporal space. While structural and content-level features represent the characteristics of individual conversations, the user-level features capture the characteristics of users who participate in simultaneous conversations. In the genetic algorithm, we assess the likeli-

hood of a user's action in a conversation based on the output of these two machine-learning models. Experimental results show that this technique can generate accurate conversation topological structures over time, and can accurately predict the volume of messages and the engagement of users over time.

We show the effectiveness of our approach on two groups of highly related communities: nine subreddits focused on crypto-currencies and 38 subreddits focused on cyber-security topics. The prediction of user involvement over different simultaneous conversations can also be used by community organizers to control the focused discussions or to promote positive community norms.

Our solution has a number of limitations. One is that in evaluating the generated conversation trees, our model arbitrarily maps the content-level features from a distribution built from training data. In an ideal scenario, we should predict the attributes of the comments (e.g., polarity, subjectivity) to draw these features accurately. Moreover, a rich set of content-level features to capture humor, adversity, emotions, etc. could be developed to improve the machine-learning models. Another limitation is that our approach tends to repeat in predicting the user interactions seen in the training data. A better approach would use information about the users who have been exposed to a message and thus may be candidates for responding. However, this true diffusion structure is hidden and inferring it is difficult [195].

Chapter 6: Simulating Twitter Activity Using Exogenous Signals[4]

Social media platforms provide a virtual space for online communication, and they often get influenced by what happens in the outside world. These external factors can be independent of what is recorded on a social media platform, and most often related to environment, policy, or culture [13]. For example, Twitter users react spontaneously during natural disasters (e.g., the 2010 Haiti Earthquake [196], Hurricane Irma [197]). There are various cultural habits that influence online communication patterns [198]. Some policy decisions such as deplatforming [199] or content moderation [200] can shape social media communication. However, the influence from these external events can not be easily measured (or decoupled from other factors), or explicitly controlled. Distilling these various external forces is key to improving the general understanding of information dissemination in social media platforms.

One of our driving hypotheses is that taking external events into consideration may result in better predictions of user activity on a social media platform. *Can one accurately generate the social media activity on a platform (for example, Twitter) using the recorded signals from other platforms? More importantly, is that doable in the context of unexpected events, when social media users both react to unexpected news in unpredictable ways and also generate news for many news outlets?* Our objective is to predict Twitter activity with the help of exogenous sources (as shown in Figure 6.1). Our predictions include when a Twitter message is posted, what it is about (topic), who tweeted the message, and who retweeted that message (as described in Section 4.2). This finer granularity of

[4]A part of this chapter was previously published in [4]. Permission is included in Appendix A.

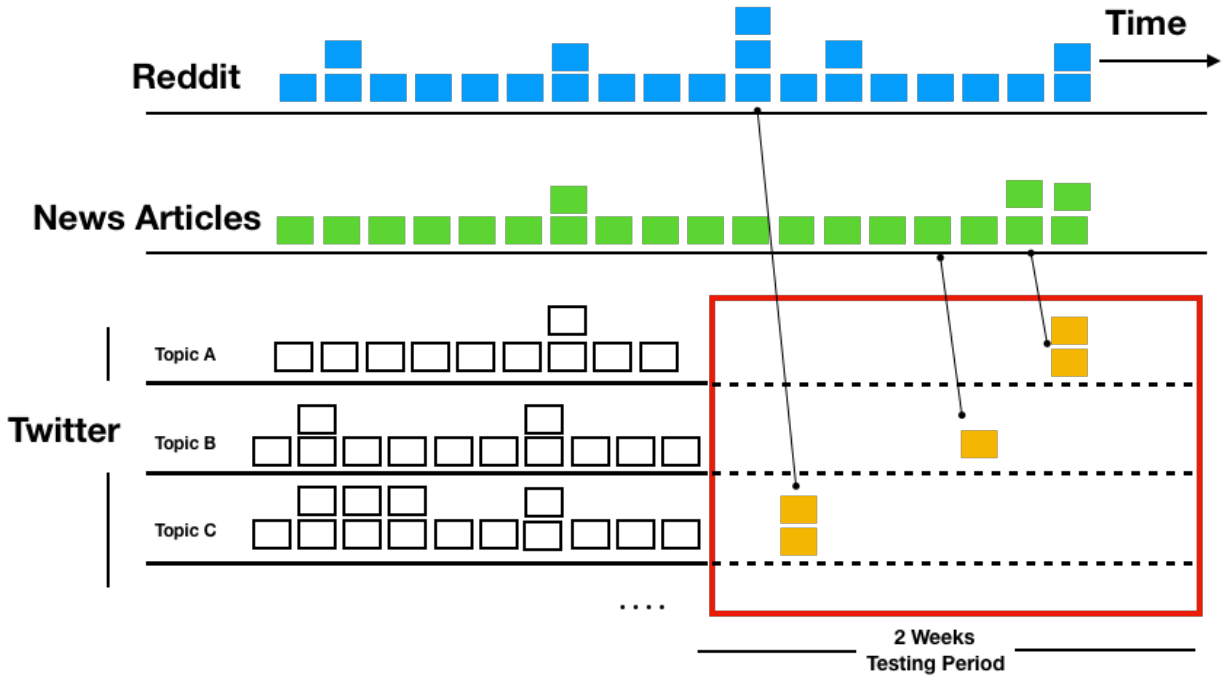


Figure 6.1: Predicting Twitter topic activity using exogenous data.

simulated activity makes the problem more difficult than simply predicting the overall daily volume of activity.

We focus on simulating an interval of two weeks at daily granularity. That means that, unlike most machine learning-based approaches that try to predict the next data point, we are predicting the activity of day d without having the ground truth of the activity on day $d-1$. We predict Twitter activity around Venezuela's latest political crisis from January 2019 to the end of February 2019. By their nature, periods of crisis do not include many repeatable events, thus it is difficult to learn and predict how social media users will react to a first refusal of international humanitarian aid, for example, or to a second round of violence against street protesters. As exogenous data, we mine news articles and the Venezuela-related subreddit, /r/vzla. The contributions of this chapter are the following:

- We present the design of a simulator that can mimic the peaks of real activity. The successful design includes modularization in order to specialize predictions to particular sub-problems, such as the prediction of the number of information cascades and the prediction of the size and growth of these cascades.
- We show that predictions using current day exogenous data work better than models that use previous day exogenous data. We push boundaries by questioning the accepted practice of using historical information from before midnight. We make the case that social media users react rapidly to news and live events, so the past is sometimes just 30 seconds ago.
- We discuss how different sources of exogenous data are beneficial for different topics that are part of the same large conversation.

This chapter is organized as follows. First, we present the simulator design and implementation in Section 6.1. Second, we describe the datasets that we use for the evaluation in Section 6.2. Third, Section 6.3 reports the simulation performance in various metrics of interest. And finally, Section 6.4 concludes with a discussion of our contributions.

6.1 Simulator Design and Implementation

Twitter users often engaged with a variety of topics over time and their reactions are often influenced by external events [192, 201]. Thus, simulating Twitter activity requires the use of signals of real-life events from exogenous data sources.

This section presents the design of a social simulator that accurately predicts Twitter activity with the help of exogenous data. Our goal is to predict two weeks of Twitter activity without any knowledge of the ground-truth Twitter activity during that period. The predicted Twitter activity

should be described by the type of action (tweet or retweet), topic, author, and the timestamp of the message.

6.1.1 Modular Design

Our design contains two modules, as shown in Figure 6.2. The Seed Module takes the historical activities in Twitter and other platforms as inputs and predicts the daily number of tweets (that we refer to as seed events) for each topic (Section 6.1.2). Second, the Cascade Module takes as input the outputs of Seed Module and generates retweet cascades in response to the seed events (Section 6.1.3). This module is similar to the solution proposed in the previous chapter (Section 5.1.1) but extends to predict Twitter information cascades. Each message in the retweet cascades is predicted with the user who posts the retweet and the day of the retweet.

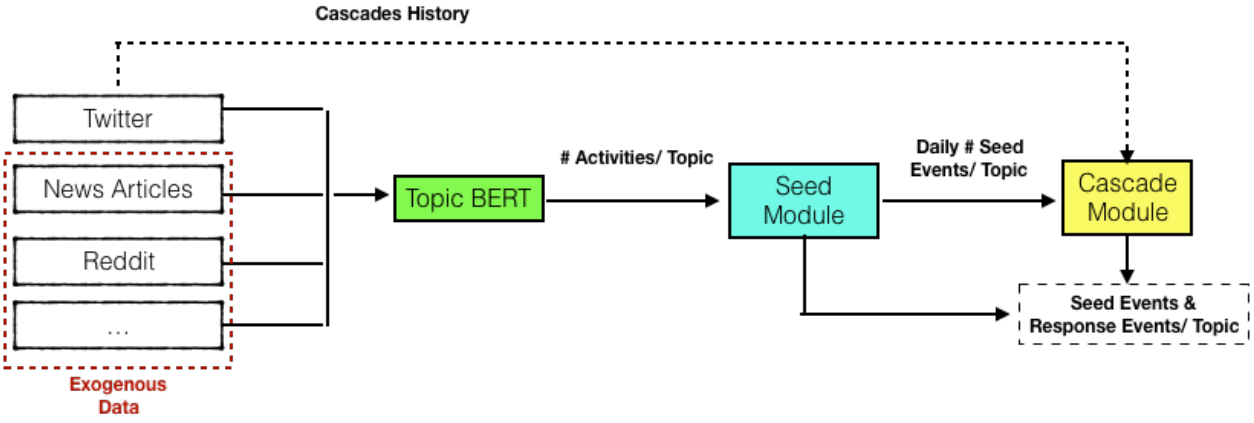


Figure 6.2: Overview of the proposed simulator design.

6.1.2 Seed Prediction Module

During our many iterations over the design of the simulator on different datasets, we noticed that correctly predicting the daily volume of tweets sets the tone for more accurate simulations. We

thus ended up designing a specialized module that is in charge of the task of predicting the number of tweets per day for the duration of the forecasting window. In addition, this module also assigns to each tweet the user who authored it.

We first attempted to predict the overall volume of tweets and then split it into topics. It turned out that accurately predicting the daily fraction of tweets that belong to a topic potentially informed or drowned out by unpredictable events such as mass protests, military interventions, or declarations of international support was challenging. We thus decided to directly predict the daily number of tweets per topic.

We implemented this module using machine learning. We trained a neural network on Twitter historical activity (expressed as a number of tweets per day) and its contemporary exogenous data signal. Given the exogenous events of day d from the forecasting window, the trained algorithm predicts the number of tweets from day $d+1$. We also experimented with using Twitter activity in day d to predict the Twitter activity of day $d+1$. This approach required us to use the predicted Twitter activity of days 2, 3, etc. of the forecasting window as input for the following day predictions since by problem definition we did not have the ground-truth Twitter data. This solution ends up compounding errors, and thus the accuracy drops over time. Moreover, this approach failed to predict bursts of activity that are evident in the ground truth. This is what prompted us to use only the volume of exogenous activity to predict Twitter activity.

For training, we built a feature vector that represents the topic as a one-hot encoding vector and the number of exogenous events related to that topic in a given day d. Exogenous features are the daily number of news articles and the daily number of Reddit messages related to the topic. Note that we can combine the exogenous sources or treat them separately. We experimented with

all versions, but for brevity, we will present experiments with only two, the news only and Reddit only. The target variable is the number of tweets related to a topic in the day $d+1$. For training, we identify the best hyperparameters using 5-fold cross-validation. We ended up with a neural network of 3 hidden layers with the sizes of 15, 10, and 5 neurons. We used the Adam optimizer and the mean squared error loss function.

We assigned users to the predicted tweets randomly with probability proportional to the user spread score [202]. Higher spread scores indicate that the identified users have more potential to spread the information. While this heuristic is not optimal, it captures the activity of influential users better than other heuristics that we considered, such as the number of tweets or the number of followers.

6.1.3 Cascade Generation Module

We use this module to generate the retweet trees (information cascades) in response to predicted tweets. We chose this approach in order to provide fine-granular predictions that can mimic user activity patterns, not only volume.

Twitter information cascades consist of a collection of retweets originated from an original tweet. While the original tweet and retweet messages are labeled with the user and timing information, the Twitter API does not provide whether the retweet is in response to another retweet [203]. Due to this limitation, we can only construct the chain of retweets from ordering the retweet messages by the timestamp. This would construct a retweet star where all the retweets are connected to the original tweet. However, this is not realistic as most often users retweet other user's retweets instead of the original tweet [180]. Different techniques are built to approximate the true retweet

tree [203]. We reconstruct the retweet tree using the time-inferred diffusion algorithm [110]. This technique connects a retweet with a previous retweet/tweet utilizing the Twitter follower network.

We use the same probabilistic approach as discussed in the previous chapter (Section 5.1.1) to generate the cascade structure with users and timestamps. After this procedure, we end up with cascade trees rooted in an original tweet in which each node has a user and a time assigned. However, we do not proceed with the optimization step used in the previous solution. This optimization step requires the output of two machine learning models that use the features for the users who posted the messages. These users appear in both training and testing data, thus allowing the model to make predictions based on their history. But we noticed that new users make up the majority of the Twitter population. Due to this limitation, we rely on a separate sub-module that predicts the daily rate of newly engaged users to inform the cascade generation process.

In this sub-module, we rely on the same feature vectors as used before in the seed module, and construct two sets of training examples depending on the exogenous source (in our case, news articles and posts on Reddit) that we select to extract features. We trained one neural network for each set of training examples, and used the trained neural networks to predict the daily number of newly engaged users for each topic. Similar to the seed module, we used the exogenous events on the day before the predictions.

We assign new users to the cascades as follows. We select leaves of the cascades predicted for each topic and assign those users a completely new and unique identifier. This approach is due to our observation that the majority of new users participate in the cascades as leaf nodes (i.e., they retweet the original tweet or another retweet, but their retweets are not reshared back). This process is repeated until we match the daily number of new users predicted.

6.2 Datasets

For the last two decades, the Venezuelan society has experienced a pervasive sociopolitical fragmentation fueled by differences of interests, identities, and politics. In Venezuela, the political spectrum is for the most part divided into two parties: Chavism, those who support the political ideology of the late president Hugo Chavez, and Anti-Chavism, those strongly opposed to Chavez's legacy. Today Chavism still maintains control of the Venezuelan political system with Nicolas Maduro as the head of state. However, failure to manage globalization, lack of investment in infrastructure, and a poor administration have put the country in the grip of a significant economic collapse. As a result, it has contributed to a significant rise in crime and violence, lack of essentials, shortages of medicines and food, and an unprecedented humanitarian crisis.

6.2.1 Venezuela Political Crisis Events

For this study, we focus on data specific to on-the-ground events in Venezuela developing in early 2019. These events highlight a period of high political tension which resulted in nationwide protests, militarized responses, and incidents of mass violence and arrests. Figure 6.3 shows a summarized timeline of the political events described below.

The 2019 Venezuelan political crisis has its roots in the controversial re-election of Nicolas Maduro as the country's president on January 10th. The event marked the beginning of a presidential crisis driven by claims of illegitimacy and reports of coercion and fraud. During the following days, the opposition-controlled National Assembly widely denounced the re-election as fraudulent and mandated an order of succession. On January 23, the opposition leader, Juan Guaidó, declared himself interim president of Venezuela in an effort to restore democracy and constitutional rights.

The event erupted widespread protests to put pressure on Maduro's administration to resign from office, and it formed a coalition of countries in support of Guaidó. In response to this, Maduro's government ordered the armed forces into the streets to maintain social order and disperse mass protests. These intense and violent clashes between the military and opposition supporters continued during the first couple of weeks of February and resulted in massive lootings, a large number of detentions, and dozens of injured.

On February 2, Guaidó announced a plan together with an international coalition to bring humanitarian aid into Venezuela on February 23. At the same time, Maduro rejected international aid offers and ordered the immediate closure of the Brazilian and Colombian borders to impede its delivery. A day before the international aid delivery, two dueling concerts took place simultaneously at the Colombia-Venezuela border. The "Aid Live" concert was organized with the purpose to help raise money and support for the international humanitarian aid effort. On the other hand, Maduro's government organised the "Hands Off Venezuela" concert to counteract the rival concert and reject aid efforts. On February 23, the plan to bring humanitarian aid into Venezuela was met with a violent standoff between military forces and those accompanying the aid. Clashes continued to run rampant over the next couple of days, and eventually, it was reported that none of the aid shipments were able to enter the country.

6.2.2 Data Collection and Processing

In this section, we describe the Twitter dataset covering the Venezuelan political crisis, exogenous data sources, and other data preprocessing and enrichment steps in detail.

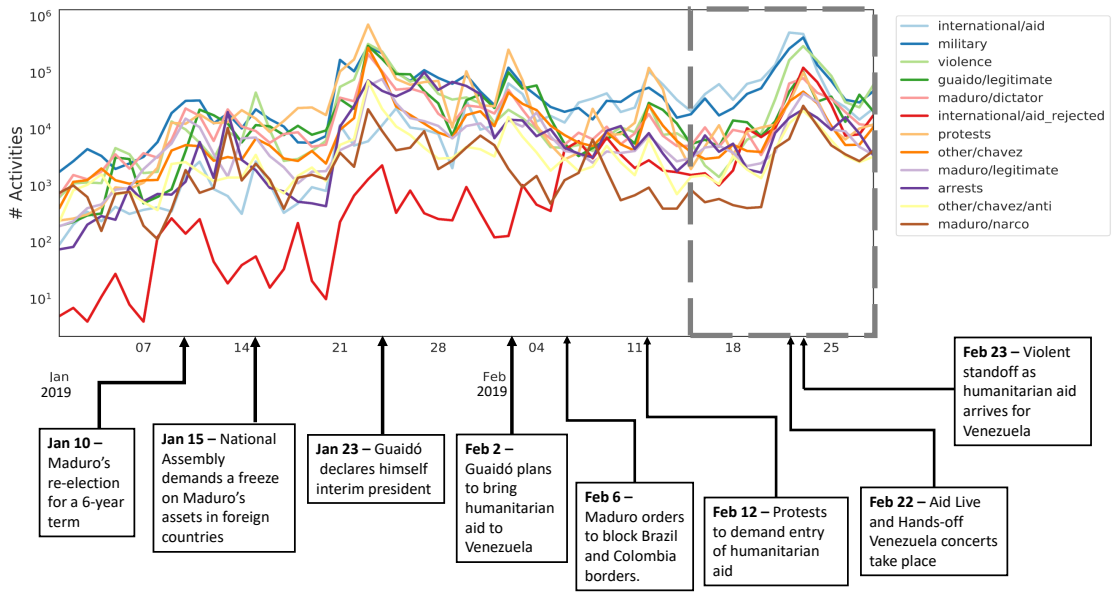


Figure 6.3: Timeline of Venezuela political events. We also show the timeseries of Twitter messages for 12 topics in our dataset. The box represents the testing period for our simulator (February 15—February 28, 2019).

6.2.2.1 Twitter Data

The Twitter data was collected over a period of two months (January 1st to February 28th, 2019) using GNIP, a data collection API tool, and based on a list of keywords relevant to the Venezuelan political crisis. Table 6.1 presents the complete list of keywords for data collection. The resulting dataset consists of 1,104,175 seed messages including tweets, replies, and quotes done by 273,392 users, and 11,681,723 retweets by 889,139 users. The majority of messages are in Spanish (86%) and English (6%). We note that user identities are anonymized in this dataset. Each Twitter record in our dataset contains the following information: an assigned unique identifier, the unique (anonymized) ID of the user who posted it, the timestamp of the message, the respective content of the message, and its type (whether a tweet, quote, reply or retweet). This dataset was provided privately as part of DARPA SocialSim program.

Table 6.1: Keywords used for data collection.

#23Ene, #23Feb, 23 de Enero, 23 de Febrero, Aid Venezuela, #BravoPueblo, Caracas, Maturin, Maracaibo, #Chavismo, #Chavistas, FANB, #FreeVenezuela, #FueraDictadura, Fuerza Venezuela, GNB, #GritemosConBrio, #GuaidoPresidente, #JGuaido, Juan Guaido, #LasCallesSonDelChavismo, Leales siempre traidores nunca, Libertad para Venezuela, Freedom for Venezuela, #VamosBien, #MaduroDictador, #MaduroUsurpador, Nicolas Maduro, #SOSVenezuela, #VenezolanosEnElMundo, Venezuela Aid Live, #WeAreMaduro, Yankee go Home, #HandsOffVenezuela, #FebreroRebelde, #NoMasDictadura, Maduro, #AbajoCadenas, Venezuela Crisis Humanitaria, Maduro Ilegitimo, Guaido, Chacao

In order to annotate Twitter messages with topics, we worked alongside three research collaborators, who are subject-matter experts regarding the Venezuelan political context. The annotators are fluent in both English and Spanish, and also familiar with particular jargon and specialized terms commonly used in Venezuela. We conducted a thorough exploration of our dataset corpus in order to identify the most representative topics originating from online social media discussions. Our initial attempt at topic assignment resulted in the identification of 10 top-level topic groups: Guaidó, Assembly, Maduro, protests, arrests, violence, international, military, crisis, and others. While some of these top-level topics express important information on their own (e.g., protests, arrests), others have no well-defined meaning or are too vague on their own (e.g., international). Hence, some top-level topics were further extended to account for more informative and detailed semantic topic groups. For example, the international topic was broken down into other sub-topics (e.g., international/aid and international/aid_rejected). These sub-topics are more focused and make explicit reference to specific on-the-ground events unfolding in Venezuela. Overall, this effort resulted in a total of 49 sub-topics. Practically, it is not feasible to manually label millions of messages. So, in order to automate the annotation process, we conducted a semi-supervised classification task consisting of two steps: (1) manually annotating an initial subset of messages, and (2) training a multilingual BERT model to classify each message with one or multiple such sub-topics.

The manual annotation process was conducted over a corpus of 11,218 messages and consisted of an 8 to 1 ratio of single-annotator annotations to all-annotator annotations. That is, for every 8 messages annotated by each annotator individually, there is one message that all annotate. Periodically, we calculated the inter-annotator agreement given by Cohen's Kappa and Fleiss' Kappa scores. This process allows us to identify and ignore topics with low reliability and quality. Particularly, we narrowed down our initial 49 topics to the following 12 topics: international/aid, military, violence, guaido/legitimate, maduro/dictator, international/aid_rejected, protests, other/chavez, maduro/legitimate, arrests, other/chavez/anti, and maduro/narco. These 12 topics reported inter-annotator agreement scores of 0.64 for the weighted average Cohen's Kappa, and 0.7 for the Fleiss' Kappa measurement. Previous work has also found similar Cohen's Kappa agreement scores in a variety of datasets [204].

After manual annotation, we trained a BERT model for topic annotation. Previous works have found great success using BERT for multilingual text classification tasks [205]. Hence, in this study, BERT is preferable since our dataset consists of a mix of multiple languages. The BERT model was trained on 10,097 unique text documents and evaluated on a 10% test set (1,121 texts). We used stratified sampling to ensure that the train and test sets have approximately the same percentage of samples of each topic class as in the original manually annotated corpus. The model obtained a precision of 67%, recall of 66%, and F1 score of 66%.

6.2.2.2 Exogenous Data

In order to evaluate the interplay between Twitter activity and the different signals from contemporary exogenous data, we collected data from Reddit as well as mined news articles relevant to the Venezuela political crisis. We collected discussions structured around one of the largest

Venezuela-related subreddits, `/r/vzla`. The subreddit Venezuela community often engages in political discussions about the political spectrum in the country, which most likely are not going to be covered in conventional news outlets. Hence, it offers a different perspective about the ongoing political crisis in Venezuela and may provide useful signals to predict online activity on other platforms. The Reddit data was collected via the publicly available Reddit API. A query against the period of January 1 to February 28, 2019 returned a total of 4,933 posts and 51,136 comments done by 3,220 users. The corresponding text content on posts and comments and the timestamp of the postings were also collected.

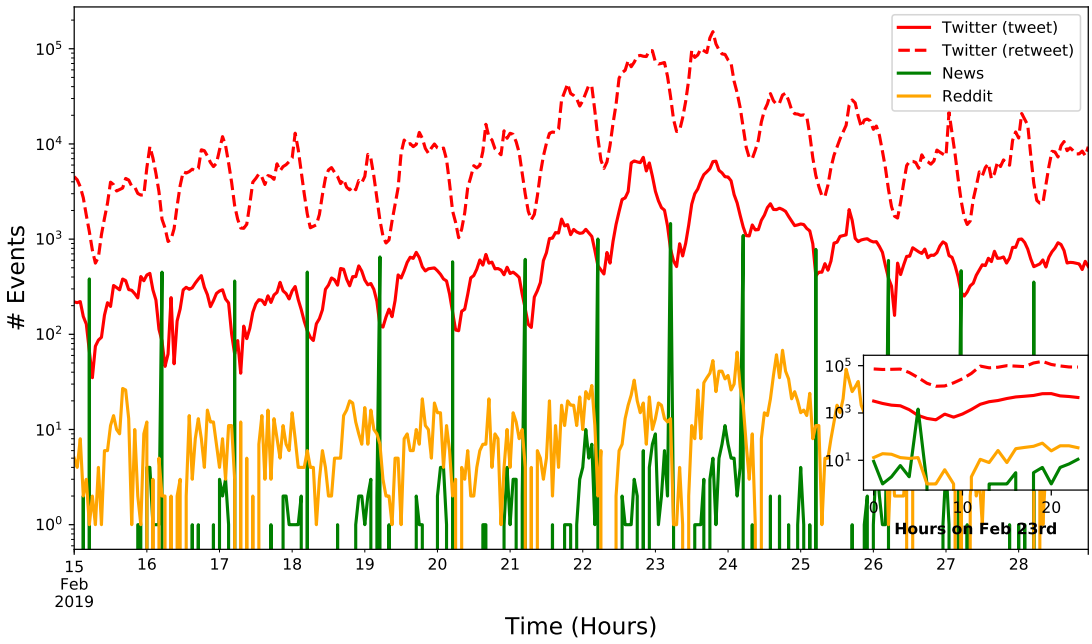


Figure 6.4: Timeseries of tweets, news articles, and Reddit messages.

The news article data was collected via a publicly available geopolitical event database, GDELT [206]. The database consists of machine-coded events extracted from news reports on a variety of news sources. The GDELT database is updated from the news articles published every 15 minutes intervals. We queried the database using the "Venezuela" search term between January 1

and February 28, 2019. This returned a total of 138,009 source URLs, where the Venezuela keyword occurs anywhere in the content of the document. We also gathered the source text for each article and the date when it was published. We used the article reference time in GDELT to retrieve the hour of publication.

We assigned topics to exogenous data by running the previously pre-trained BERT model over the content of news articles and Reddit messages. We randomly sampled 500 Reddit posts and news articles to measure the reliability of this classification. We were only able to find 5% false positives. This results in a total of 2,021 posts and 31,295 comments on Reddit, and 81,887 news articles to be associated with at least one topic of interest. Figure 6.4 shows the hourly activities of Twitter and respective exogenous sources.

6.3 Evaluation

We measured the accuracy of the generated Twitter activity per topic by comparing it against ground truth and against two baselines. We report performance using three metrics: (i) the daily activity volume as represented by the number of tweets and retweets, (ii) the number of newly engaged users every day, and (iii) the page rank distribution of the user interaction network.

Due to the complexity of our prediction problem (e.g., who responds to whom in which topic, and when), comparing our solution with other related work is not straightforward. We compare our solution with two baselines extracted from training data. The first baseline, Replay, simply repeats all events from the past two weeks. Thus, the only change is in the timestamps of the events and there are no new users (since every user was also active exactly two weeks before in the same topics). The second baseline, Sampling, draws full Twitter cascades at random to match the average

daily volume of activity per topic observed in the last two weeks of training data. Thus, while this sampling ignores the variations in volume from one day to another (see the corresponding flat line in Figure 6.6), it aims to approximate the overall volume of shares over a 2-week period. Because these baseline models repeat events from the recent past, they are very challenging to outperform in simulating user activities in multiple social platforms, as shown in [107].

6.3.1 Predicting the Number of Shares

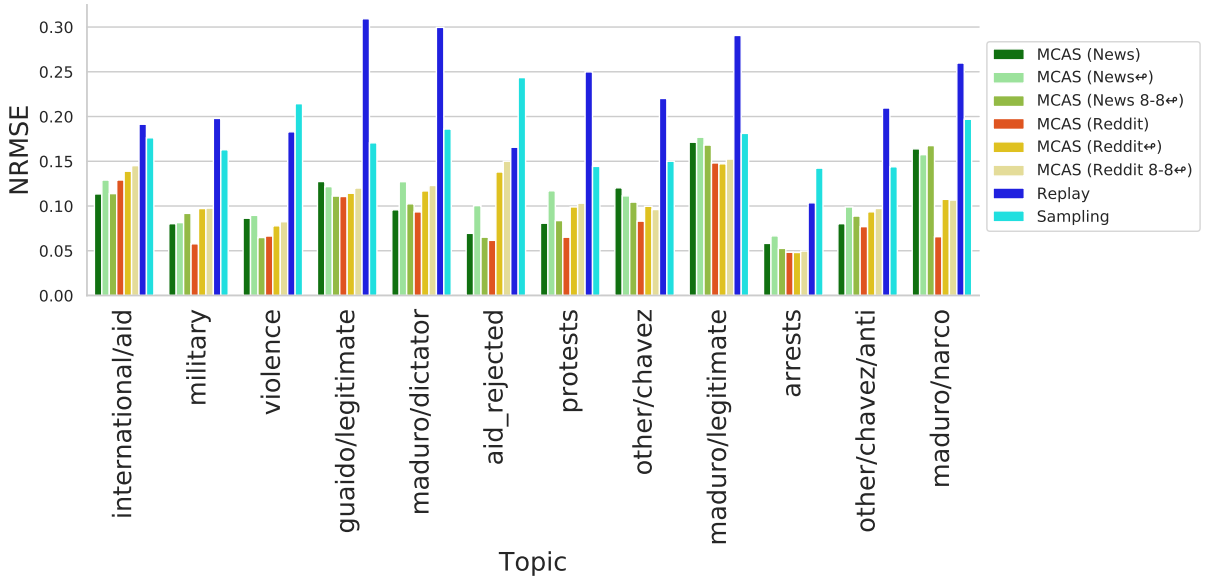
Predicting the number of shares for each topic is challenging because of burstiness and because different topics dominate at different times [148]. For example, the two big spikes in Twitter activities during the last two weeks of February (Figure 6.3) are mainly due to the Venezuelan Aid Live concert on February 22, and the violent standoff between military forces and those accompanying the aid and protesting against the regime on February 23. While the spike on February 22 was dominated by activities related to *international/aid* topic, the spike on February 23 was due to three topics popular that day: *military*, *violence*, and *protests*. This is where exogenous data (especially extracted from news reports) can be valuable for capturing the variations in popularity of Twitter topics over time.

We evaluated our solution on predicting the number of tweets per day per topic during the forecasting period. Figure 6.4 shows that the GDELT exogenous data precedes Twitter (and Reddit) activity. This observation suggests, on one hand, that Twitter reacts very quickly to the peak of news as recorded in GDELT; on the other hand, there are few updates after about 8 am in GDELT. Reddit discussions are quite spread out over the day, but many comments in our dataset are posted in late afternoon.

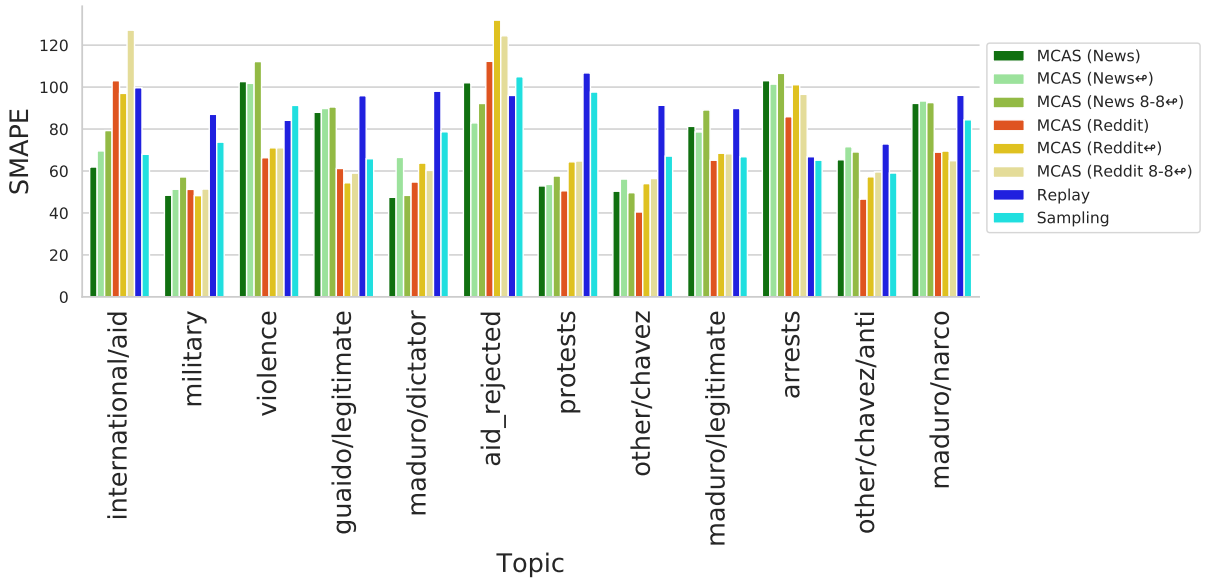
We thus tested three scenarios corresponding to the time of the exogenous data. First, exogenous features are from the day of predictions. This scenario is realistic in the context of filling in gaps of data from the past (thus, the events as recorded in exogenous data already happened). The second scenario assumes that the exogenous data are from the previous day. This scenario holds even for the context in which our Twitter activity generator predicts the future based on a recent past. The third scenario corresponds to the assumption that exogenous data is from the past, but we roll the day from 8 am to 8 am, in order to catch the peak of activity in GDELT, thus challenging the common practice of delimiting days at midnight. This scenario allows for a shorter delay between the GDELT peak of events and the start of the daily Twitter activity patterns, yet is realistic for both predicting future activities and generating past activity. For each scenario we extracted the corresponding features and trained three neural networks.

We report the accuracy of predicted daily volume of activity by two metrics: normalized root mean squared error (NRMSE) and symmetric mean absolute percentage error (SMAPE). For NRMSE, we take the normalized cumulative values in the prediction and ground truth vectors to calculate the root mean squared error. While NRMSE is scale-independent and evaluates the temporal patterns of two time series, SMAPE accounts for the scale of the error.

Figures 6.5a and 6.5b show the performance of predicting the volume of tweets over time for the 12 topics. We note the following. First, multiple variants of our solution capture the trend of the number of tweets closer to the ground truth than any baselines for most of the topics. As reported in Figure 6.5a, all the variants of our solution perform better than the baselines in NRMSE for all topics. More importantly, we predict the big spikes in the number of tweets for most of the popular topics (as shown in Figures 6.6a and 6.6b). While the Replay baseline predicts some spikes,



(a) Daily pattern of tweets (NRMSE)



(b) Daily volume of tweets (SMAPE)

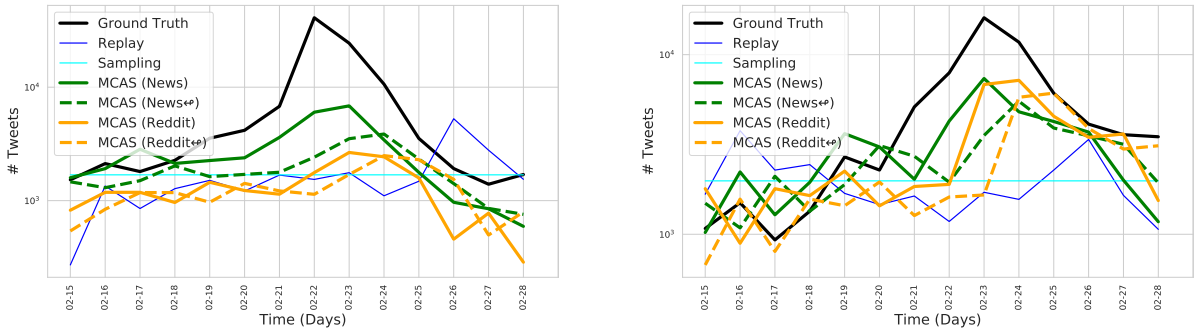
Figure 6.5: Model performance of predicting tweets over time. We also report the performance of MCAS (News 8-8↩) and MCAS (Reddit 8-8↩) that use the respective exogenous features from the last 24 hours before 8 a.m. each day to predict the tweets in the next 24 hours.

they are not timed similarly with the spikes in the ground truth, clearly making the point that *contemporary* exogenous data is necessary for accurate forecasting.

Second, our solution fares better than the baselines not only in trend, but also in the actual volume of tweets over time. As shown in Figure 6.5b, our solutions perform better for all topics except for *arrests*. For the most popular topic (*international/aid*), the minimum SMAPE is 61, while Replay baseline records a SMAPE value of 99. Our worst performance is for the *arrests* topic, where our solution failed to accurately predict the scale of multiple spikes. We noticed that the spikes in *arrests* are not correlated between Twitter and other exogenous platforms. This might be due to the emotional response in Twitter for the discussions related to the *arrests* topic that might not be timely captured in the news articles or Reddit messages.

Third, out of the two models that use the current day exogenous features, the model that used only Reddit features predicts the trend of tweets better than the model that used only news features in all topics except for *international/aid* (as shown in Figure 6.5a). We noticed the online community in Reddit who often engage in political discussions may provide a different perspective about the on-going political crisis than what is usually covered in the news articles. These external signals that originated from Reddit are helpful to predict the trends of activities on Twitter for many topics. The exception in the performance of the topic *international/aid* might be due to the timely coverage of the humanitarian aid effort in news articles.

Fourth, as expected, using current day exogenous data leads to more accurate predictions than using the previous day exogenous data. As shown in Figures 6.5a and 6.5b, the model that uses the current day exogenous features records lower NRMSE and SMAPE values than the models that used the previous day exogenous features for most of the topics. We noticed the models that



(a) international/aid

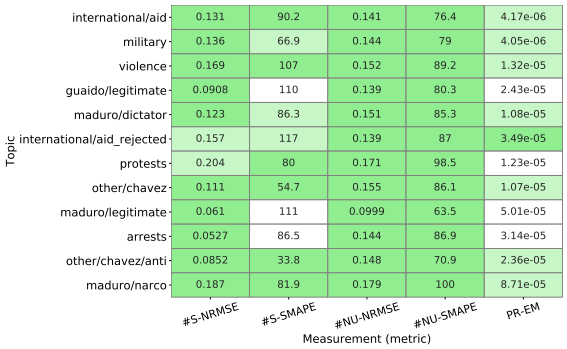
(b) military

Figure 6.6: The number of tweets per topic. MCAS (News) and MCAS (Reddit) models use the respective exogenous features from the day of predictions, while MCAS (News↩) and MCAS (Reddit↩) models use the respective exogenous features from the day before the predictions. We only visualize the time series for the two most popular topics to reduce visualization clutter.

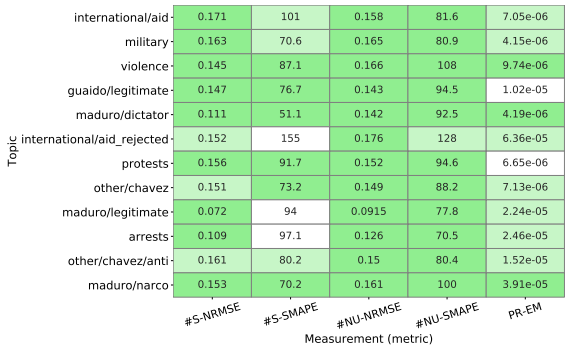
use the news articles in the last 24 hours before 8 a.m. perform better on predicting the trend of tweets than the models that use the news articles in the previous day of predictions (as shown in Figure 6.5a). This might be due to the particularity of the GDELT database, where the majority of news records were published around 5-6 am (as shown in Figure 6.4).

We also compare the volume of total shares on the same metrics between the prediction and ground truth. While the seed module (Section 6.1.2) is responsible for predicting the number of tweets and thus the number of information cascades, the size and growth of these cascades (i.e., retweets over time) are predicted by the cascade module (Section 6.1.3). To maintain consistency across our different modules, we also generate retweets for the tweets predicted by different variants of the seed module, but only report the performance for four variants to reduce the visual clutter. Figures 6.7a, 6.7b, and 6.7c show the performance of predicting the volume of total tweets and retweets for the 12 topics. We have two main observations.

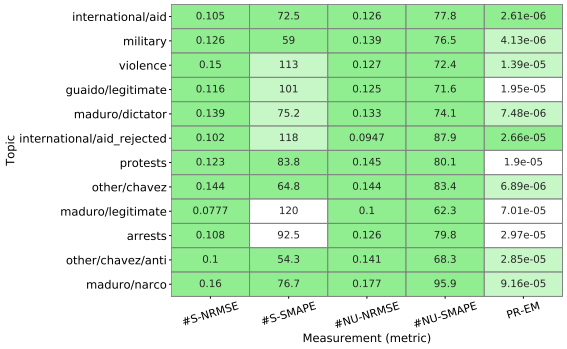
First, similar to the performance of the seed module, the cascade module also captures the trend of number of shares closer to the ground truth than any baselines for most of the topics. We



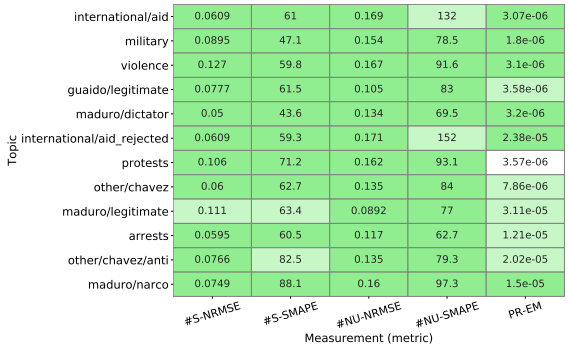
(a) MCAS (News↩ρ)



(b) MCAS (Reddit↩ρ)



(c) MCAS (News 8-8↩ρ)

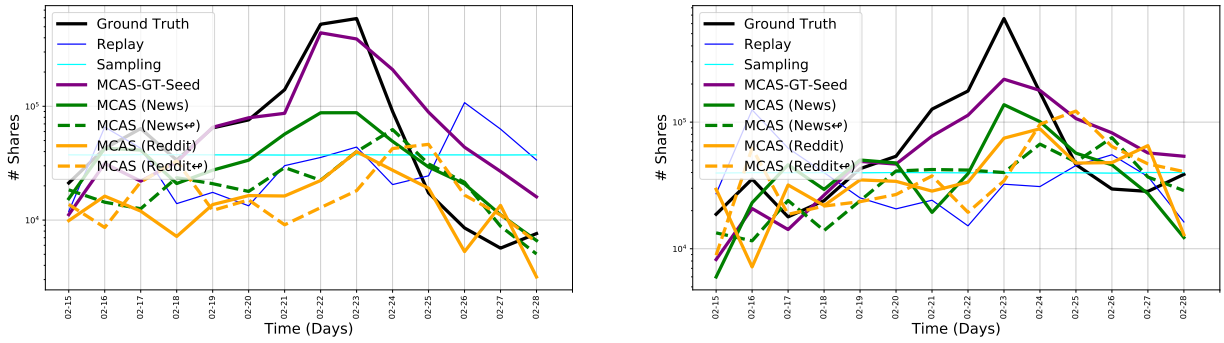


(d) MCAS-GT-Seed

Figure 6.7: Overview of the accuracy in forecasting Twitter activity. We report performance in five metrics (shown along the x axis) after compared with ground-truth data: daily number of tweets/retweets over time (#S) as measured by NRMSE and SMAPE, daily number of newly engaged users over time (#NU) as measured by NRMSE and SMAPE, and page rank distribution (PR) of the user interaction network as measured by Earth Movers (EM) distance metric. The colors of the cell represent comparison with the baselines: the darkest shows better performance than both baselines, the non-colored shows lower performance than both baselines. MCAS (News 8-8↩ρ) uses the respective exogenous features from the last 24 hours before 8 a.m. each day to predict the tweets in the next 24 hours. MCAS-GT-Seed uses the ground truth tweets to generate retweets.

noticed the temporal pattern of total shares is driven mostly by the temporal pattern of tweets predicted by the seed module.

Second, our solution captures the volume of shares over time for most popular topics (as shown in Figures 6.8a and 6.8b) except *international/aid-rejected*, *maduro/legitimate* and *arrests* (as shown in Figures 6.7a and 6.7b). While we predict the number of tweets using a learning



(a) international/aid

(b) military

Figure 6.8: The number of shares (tweets and retweets) per topic. MCAS (News) and MCAS (Reddit) models use the predicted tweets from the respective exogenous features, and MCAS-GT-Seed uses the the ground truth tweets to generate retweets. We only visualize the time series for the two most popular topics due to space constraints.

algorithm, the number of retweets is the output of a generative algorithm. Thus, we expected that prediction errors in the seed module carry over to the predictions in the cascade module. For example, the lowest performing topics in the seed module (*international/aid_rejected*, *arrests*) is the lowest performing in the cascade module as well.

In order to separate the errors from the seed module (predicting tweets) and the cascade module (predicting retweets based on input tweets), we include in our evaluation the unrealistic scenario when the tweets from the ground truth data are given as input to the cascade module. As expected, we predict the volume of retweets more accurately when the cascade module used the ground truth tweets (as shown in Figure 6.7d).

6.3.2 Predicting User Engagement

While our solution assigns a user identity (whether previously seen in the training data or not) to each tweet and retweet, we do not attempt to predict exactly what user will post when and on what topic. However, we evaluate the accuracy of user assignments by comparing the rate of

newly engaged users, and the user interaction with the ground truth. This is particularly relevant for application scenarios such as designing network intervention techniques, when one would like to investigate "what if" scenarios (e.g., blocking some user accounts) and their consequences at particular times.

First, we report the predicted number of new users per day compared with ground truth data in the same two metrics as before, NRMSE and SMAPE. Figures 6.7a and 6.7b show that our models outperform the respective baselines across all 12 topics with respect to NRMSE and SMAPE. We also found that the performance of our solutions varies based on the topic of interest. For instance, models using only Reddit features show better performance than those using only news in *other/chavez* and *maduro/narco* topics, as shown in Figure 6.7b. On the other hand, models using news features seem to do slightly better in the *violence* and *international/aid_rejected* topics. This suggests that different exogenous sources offer unique signals that are helpful for a particular set of topics, but not all. For example, the Venezuela subreddit tends to engage more in discussions expressing dissatisfaction towards the current government, which may not necessarily be reported in the news articles. Hence, Reddit features could potentially be more valuable and stronger predictors than news features for those topics expressing signs of discontent with the Venezuelan government.

Second, we are interested in comparing the user interaction networks of the predicted activity and the ground truth data, focusing again on each topic independently. For this, we split the events in the prediction and ground truth dataset by topic. Then, we create a directed retweet network for each topic in which an edge points from the user who retweeted to the user who posted the tweet. Finally, we calculate the page rank distributions, and compare the predicted distributions and ground truth distributions using Earth Movers (EM) metric distance [207].

We report the performance of network structural measurements in Figure 6.7. Our observations are the following. The page rank distribution of the user interaction networks is closer to the ground truth than the *Sampling* baseline method for a majority of topics (as shown in Figures 6.7a, and 6.7b). For example, our solution records the lowest EM distance values in the three most popular topic networks compared to both Sampling and Replay baseline (Figures 6.7a).

We also learnt that the network structures predicted by the Replay baseline model are hard to beat in this network measurement. As we generate cascades starting from the seed user positions in the user interaction network, the correct seed user assignments matter in our solution to predict the network structure more accurately. To understand the impact of seed user assignments, we run the cascade module for the ground truth seeds with the correct authorship information. For a fair comparison, we keep the same cascade parameters across all solutions. This solution accurately predicts the page rank distribution for most of the topics (as shown in 6.7d). Since we randomly select users previously seen in training data, we only predict the long-lived users as seed users who tend to be more influential in the forecasting period. We believe future improvements on seed user assignments will improve the overall results on network structural measurements.

6.4 Summary and Discussion

This chapter presents the design and evaluation of a simulator capable of generating realistic Twitter activity during intense real-world events that lead to peaks of activity and a changing mixture of popular topics. The simulator uses Twitter data and data from exogenous sources (such as Reddit and news articles as recorded in GDELT) for training, and produces Twitter activities (with details on which user tweeted or retweeted when and on what topic) over two weeks during which only

contemporary exogenous data is available to the simulator. We show on real data collected during the Venezuela political crisis from January-February 2019 that our simulator generates activities that follow the ground truth timeseries per topic in terms of message volume and user engagement. Our code is available for download [194].

Various observations from our effort on this problem may be relevant to researchers focused on related topics. First, taking into account exogenous data is necessary for simulating the activity of some social media platforms, especially Twitter. What sources of exogenous data are most representative depends on the topics of interest. For example, we discovered that Reddit conversations more accurately predicted the Twitter activity related to the late president of Venezuela, who is understandably rarely mentioned in the news. Similarly, Reddit discussions about arrests mirror better the corresponding Twitter discussions than news articles do, perhaps because they are emotion-charged. It is possible that taking semantics into account will improve the ability of forecasting some of the more challenging topics [208, 209].

Second, peaks of activity are difficult to predict. While our solution got the timing of the peaks right for many topics, we sometimes failed to predict the correct volume. Predicting when the volume of activity for a topic peaks can have many applications, such as identifying the "Pump and Dump" group activity in crypto-currency.

Third, we showed that, given the very short reaction time to real-life events, it is important for researchers to re-evaluate what "the past" means. In particular, restricting exogenous data to the day previous to the one whose activity is to be predicted is unnecessarily limiting. The "past" on Twitter is only a few minutes ago. Depending on the time granularity of the predictions sought, a smaller gap between when the exogenous events took place and the Twitter activity to forecast is

preferable. We showed there are significant improvements in forecasting accuracy based on previous day exogenous data when the previous day is shifted to capture the peaks of activity that are likely to set the tone for the next day.

Fourth, we reached this modular design after many trials experimented over different case studies. In particular, we experimented in the past with end-to-end machine learning algorithms, including long-short term memory approaches to better capture trends over time. In our experience, end-to-end solutions will find the middle ground in the multitude of performance metrics it has to satisfy, but miss exceptional cases (such as peaks of activity or peaks of new users engagement). A modular design allows for optimization of the most important dimensions of the simulated data (such as timing or number of tweets) and can also allow for corrections of unlikely outcomes (such as more users are predicted to tweet than the number of predicted tweets in some time interval).

Our solution has a number of limitations, some by choice and some related to the results of our evaluation. We chose not to simulate all types of Twitter actions (such as quoted retweets and replies) because they make up a small percentage (0.8%) of the total Twitter activity volume. We also chose to ignore semantic information in order to see how much we can push a general simulator that might be applicable later to different extractions of topics, and different types of crisis events. We believe semantic information can contribute to such a simulator and we are interested in addressing this in future work.

Chapter 7: Conclusions and Future Work

Social media data is useful to understand various properties of online communication including polarization [4], influence operations [210], and cross-platform information dissemination [211]. This dissertation contributes to two studies on social network data: protecting the privacy of individuals in publicly available social network data, and simulating online user activity in various social media platforms.

We proposed a data-driven framework that identifies the relationships between graph vulnerability and graph properties. Specifically, in Chapter 2, we introduced a framework that provides a quantification of graph vulnerability as measured by the success of a machine-learning based re-identification attack. This framework provides mechanisms to explain the relationship between graph vulnerability and graph characteristics. Our study shows that protecting graph privacy is harder than previously considered [10, 11]. For example, previous studies show that preserving the degree distribution or the degree correlation increases graph vulnerability [41] and thus disturbing them is a necessary condition for graph anonymization. We show that preserving other network properties independent of the degree distribution can reveal node identity as well.

In Chapter 3, we improved this framework to measure the cost of graph vulnerability imposed by the attributes of a labeled graph. We show that the addition of even a single binary attribute to nodes in a network increases the chance of revealing node identity. Our empirical results show that graph vulnerability depends on the population diversity with respect to the attributes considered,

but does not depend on the placement of such attributes biased towards homophily. This improved understating can guide the data practitioner in selecting anonymization techniques that provide the appropriate tradeoff between utility and privacy. We make this framework publicly available [42].

The second part of this dissertation contributes to the development of social simulators that predict social media activity. Specifically, Chapter 4 - 6 present the design of a simulator for a particular platform that generates realistic user activities.

We proposed a modular design of a simulator to predict finer granular social media activity. Chapter 5 presents one part of this modular design that generates conversation structures with user and timing information. We show that the properties of a pool of conversations can be predicted given only a group of original posts without relying on the initial reactions in the same conversations [115]. Our methods include machine learning algorithms that help to assess the goodness of the generated conversations with respect to the authorship, timing and structure of a conversation. Our code is available for download [194].

This solution had two main limitations. First, the model requires the original post information to predict the remainder of a conversation. In an ideal scenario, the simulator would not have any ground truth information in the testing period. Second, the model can not make predictions for newly engaged users. In social media platforms like Twitter, a majority of users engaged in discussions are new (that is, not seen in the past engaged with the same topic of interest).

Chapter 6 presents the overall simulator design that is built from the cascade solution. We build specialized modules to overcome the limitations presented in the previous solution. First, we develop a specialized module to predict the original post information. Specifically, we use exogenous features (such as news articles and posts on Reddit) to predict tweets information (i.e., original posts

on Twitter). Second, we predict when new users join the discussion by predicting the daily timeseries of new user engagement. These new users are a subset of the inactive population in social media discussions, and low degree nodes of the interaction network. These predictions are important when assigning users to the generated cascades.

We decomposed the simulation problem into various subproblems. For example, we predict the daily volume of social media discussions (platform-level) per topic (content-level) and distribute the activity into different user populations (old and new users). Another approach would be to directly predict the activity streams of individual users, which can be used to estimate the volume of social media discussions. However, this approach fails when there are millions of users with different activity patterns (e.g., sparse, bursty, persistent, etc.).

There are various ways that we can improve the performance of a social media activity simulator. First, there are errors propagated over different modules in a pipeline design. For example, any error on predicting the volume of discussions can not be resolved later in the pipeline, as errors are getting accumulated over different modules. Accurately identifying which module penalizes overall prediction is important to make improvements. *How does the improvement made on the volume predictor impact the network structure predictions? What is the impact of predicting the rate of new user engagements in predicting the overall volume of discussions?* We need to test the modules independently under different conditions (e.g., a variety of social media datasets, simulation scenarios, etc.) to check their robustness.

Second, we need to evaluate the usefulness of exogenous features across multiple case studies. *How reliable are the exogenous data sources for predicting the popularity of social media topics in various social contexts ranging from organic discussions to discussions originated as a part of*

propaganda, influence campaigns? In this line, one can question the reliability of news articles for predicting social media activity. For example, mainstream and alternative news articles are shown to promote different topics in a disinformation campaign [212, 210], or the news articles may be censored as a part of authoritarian propaganda [213]. While we believe the modular design proposed in this work is generalizable for Twitter activity prediction, we might need to reevaluate the exogenous features across multiple case studies. There might be new exogenous data sources that would be more useful in particular cases.

Another direction of future work is to find explanations for the simulator performance. *What characteristics of the data determine the models' performance?* During our performance analysis, we have seen the simulator performing differently on different topics. This could be partly due to the influence of external events on the activity of particular topics, or partly due to the regular patterns observed in the data. For example, we have seen how Reddit messages provide different perspectives about the on-going Venezuelan political crisis than usually covered in news articles. On the other hand, there are new patterns emerging in the data that test the generalizability of simulators. These models learn to simulate according to the way that they have seen the past world through different data representations. For example, there is a big activity spike in Venezuelan social media on February 22 due to the humanitarian aid concert. Our simulator has not seen such a spike in the training data for this particular topic yet manages to predict the spike given the features extracted from the exogenous sources. In this example, the model interprets the world as seen from the lens of news articles. This suggests that the model has the ability to learn the fluctuation of social media activity relative to the exogenous activity. Further work is needed to understand the data characteristics that can explain the performance of a data-driven simulator. This improved

understanding would tell us how the models will perform in the future just by looking at data, but before training any models.

References

[1] David A Broniatowski, Michael J Paul, and Mark Dredze. National and local influenza surveillance through twitter: an analysis of the 2012-2013 influenza epidemic. *PloS one*, 8(12):e83672, 2013.

[2] Jeremy Blackburn, Nicolas Kourtellis, John Skvoretz, Matei Ripeanu, and Adriana Iamnitchi. Cheating in Online Games: A Social Network Perspective. *ACM Transactions on Internet Technology (TOIT) - Special Issue on Foundations of Social Computing*, 13(3), May 2014.

[3] David Lazer, Alex Pentland, Lada Adamic, Sinan Aral, Albert-Laszlo Barabasi, Devon Brewer, Nicholas Christakis, Noshir Contractor, James Fowler, Myron Gutmann, et al. Social science. computational social science. *Science (New York, NY)*, 323(5915):721–723, 2009.

[4] Sameera Horawalavithana, Kin Wai NG, and Adriana Iamnitchi. Drivers of polarized discussions on twitter during venezuela political crisis. In *The 13th ACM Conference on Web Science*. ACM, 2021.

[5] Arvind Narayanan, Elaine Shi, and Benjamin IP Rubinstein. Link prediction by de-anonymization: How we won the kaggle social network challenge. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 1825–1834. IEEE, 2011.

[6] Mudhakar Srivatsa and Mike Hicks. Deanonymizing mobility traces: Using social network as a side-channel. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 628–637. ACM, 2012.

[7] Shouling Ji, Weiqing Li, Mudhakar Srivatsa, Jing Selena He, and Raheem Beyah. Structure based data de-anonymization of social networks and mobility traces. In *International Conference on Information Security*, pages 237–254. Springer, 2014.

[8] Nitish Korula and Silvio Lattanzi. An efficient reconciliation algorithm for social networks. *Proceedings of the VLDB Endowment*, 7(5):377–388, 2014.

[9] Kumar Sharad. True friends let you down: Benchmarking social graph anonymization schemes. In *Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security*, AISec '16, pages 93–104, New York, NY, USA, 2016. ACM.

[10] Alina Campan, Yasmeen Alufaisan, and Traian Marius Truta. Preserving communities in anonymized social networks. *Trans. Data Privacy*, 8(1):55–87, December 2015.

[11] Michael Hay, Gerome Miklau, David Jensen, Don Towsley, and Philipp Weis. Resisting structural re-identification in anonymized social networks. *Proceedings of the VLDB Endowment*, 1(1):102–114, 2008.

[12] Chengcheng Shao, Giovanni Luca Ciampaglia, Onur Varol, Kai-Cheng Yang, Alessandro Flammini, and Filippo Menczer. The spread of low-credibility content by social bots. *Nature communications*, 9(1):1–9, 2018.

[13] Shaomei Wu. The dynamics of information diffusion on on-line social networks. 2013.

[14] Zi-Ke Zhang, Chuang Liu, Xiu-Xiu Zhan, Xin Lu, Chu-Xu Zhang, and Yi-Cheng Zhang. Dynamics of information diffusion and its applications on complex networks. *Physics Reports*, 651:1–34, 2016.

[15] Fan Zhou, Xovee Xu, Goce Trajcevski, and Kunpeng Zhang. A survey of information cascade analysis: Models, predictions, and recent advances. *ACM Computing Surveys (CSUR)*, 54(2):1–36, 2021.

[16] Narimene Dakiche, Fatima Benbouzid-Si Tayeb, Yahya Slimani, and Karima Benatchba. Tracking community evolution in social networks: A survey. *Information Processing & Management*, 56(3):1084–1102, 2019.

[17] Antonis Matakos, Evimaria Terzi, and Panayiotis Tsaparas. Measuring and moderating opinion polarization in social networks. *Data Mining and Knowledge Discovery*, 31(5):1480–1505, 2017.

[18] Jooho Kim and Makarand Hastak. Social network analysis: Characteristics of online social networks after a disaster. *International Journal of Information Management*, 38(1):86–96, 2018.

[19] Gertrude R Gauthier, Jeffrey A Smith, Catherine García, Marc A Garcia, and Patricia A Thomas. Exacerbating inequalities: social networks, racial/ethnic disparities, and the covid-19 pandemic in the united states. *The Journals of Gerontology: Series B*, 76(3):e88–e92, 2021.

[20] Huijie Lin, Jia Jia, Jiezhong Qiu, Yongfeng Zhang, Guangyao Shen, Lexing Xie, Jie Tang, Ling Feng, and Tat-Seng Chua. Detecting stress based on social interactions in social networks. *IEEE Transactions on Knowledge and Data Engineering*, 29(9):1820–1833, 2017.

[21] Dan Swinhoe. The 15 biggest data breaches of the 21st century, Jan 2021.

[22] Shouling Ji, Prateek Mittal, and Raheem Beyah. Graph data anonymization, de-anonymization attacks, and de-anonymizability quantification: A survey. *IEEE Communications Surveys & Tutorials*, 2016.

[23] Orange. Orange telecom data for development challenge (d4d), 2006.

[24] Kumar Sharad and George Danezis. An automated social graph de-anonymization technique. In *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, page 47, 2013=4.

[25] Charu C Aggarwal. On k-anonymity and the curse of dimensionality. In *Proceedings of the 31st international conference on Very large data bases*, pages 901–909. VLDB Endowment, 2005.

[26] Justin Cheng, Jon Kleinberg, Jure Leskovec, David Liben-Nowell, Bogdan State, Karthik Subbian, and Lada Adamic. Do diffusion protocols govern cascade growth? In *Proceedings of the 12th International AAAI Conference on Web and Social Media (ICWSM 2018)*, 2018.

[27] Xiang Zuo, Clayton Gandy, John Skvoretz, and Adriana Iamnitchi. Bad Apples Spoil the Fun: Quantifying Cheating in Online Gaming. In *Proceedings of 10th International AAAI Conference on Web and Social Media (ICWSM'16)*, Cologne, Germany, May 2016.

[28] Soroush Vosoughi, Deb Roy, and Sinan Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018.

[29] Shaomei Wu, Jake M Hofman, Winter A Mason, and Duncan J Watts. Who says what to whom on twitter. In *Proceedings of the 20th international conference on World wide web*, pages 705–714, 2011.

[30] Sameera Horawalavithana, Juan G Arroyo Flores, John Skvoretz, and Adriana Iamnitchi. Behind the mask: Understanding the structural forces that make social graphs vulnerable to deanonymization. *IEEE Transactions on Computational Social Systems*, 6(6):1343–1356, 2019.

[31] Sameera Horawalavithana, Clayton Gandy, Juan Arroyo Flores, John Skvoretz, and Adriana Iamnitchi. Diversity, homophily and the risk of node re-identification in labeled social graphs. In *International Conference on Complex Networks and their Applications*, pages 400–411. Springer, 2018.

[32] Sameera Horawalavithana, Juan Arroyo Flores, John Skvoretz, and Adriana Iamnitchi. The risk of node re-identification in labeled social graphs. *Applied Network Science*, 4(1):1–20, 2019.

[33] A face is exposed for aol searcher no. 4417749. http://www.nytimes.com/2006/08/09/technology/09aol.html, 2006. Accessed: 2017-07-15.

[34] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pages 111–125. IEEE, 2008.

[35] Michael Barbaro and Tom Zeller. A face is exposed for aol searcher no. 4417749, Aug 2006.

[36] Elinor Mills. Aol sued over web search data release, Sep 2006.

[37] Ryan Singel. Netflix cancels recommendation contest after privacy lawsuit, Jun 2017.

[38] Charu C Aggarwal, Yao Li, and S Yu Philip. On the hardness of graph anonymization. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 1002–1007. IEEE, 2011.

[39] Shouling Ji, Weiqing Li, Mudhakar Srivatsa, and Raheem Beyah. Structural data de-anonymization: Theory and practice. *IEEE/ACM Transactions on Networking*, 24(6):3523–3536, 2016.

[40] Shouling Ji, Weiqing Li, Prateek Mittal, Xin Hu, and Raheem A Beyah. Secgraph: A uniform and open-source evaluation system for graph data anonymization and de-anonymization. In *USENIX Security Symposium*, pages 303–318, 2015.

[41] Alessandra Sala, Xiaohan Zhao, Christo Wilson, Haitao Zheng, and Ben Y Zhao. Sharing graphs using differentially private graph models. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 81–98. ACM, 2011.

[42] Sameera Horawalavithana. Graph unmasking. https://github.com/SamTube405/Graph_Unmasking, 2018.

[43] Kumar Sharad. *Learning to de-anonymize social networks*. PhD thesis, Computer Laboratory, University of Cambridge, 2016.

[44] Reza Shokri. Privacy games: Optimal user-centric data obfuscation. *Proceedings on Privacy Enhancing Technologies*, 2015(2):299–315, 2015.

[45] Shouling Ji, Weiqing Li, Shukun Yang, Prateek Mittal, and Raheem Beyah. On the relative de-anonymizability of graph data: Quantification and evaluation. In *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*, pages 1–9. IEEE, 2016.

[46] Wei-Han Lee, Changchang Liu, Shouling Ji, Prateek Mittal, and Ruby B. Lee. Quantification of de-anonymization risks in social networks. In *ICISSP*, 2017.

[47] Arvind Narayanan and Vitaly Shmatikov. De-anonymizing social networks. In *Security and Privacy, 2009 30th IEEE Symposium on*, pages 173–187. IEEE, 2009.

[48] Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *Proceedings of the 16th international conference on World Wide Web*, pages 181–190. ACM, 2007.

[49] Shouling Ji, Weiqing Li, Mudhakar Srivatsa, Jing Selena He, and Raheem Beyah. General graph data de-anonymization: From mobility traces to social networks. *ACM Transactions on Information and System Security (TISSEC)*, 18(4):12, 2016.

[50] Pedram Pedarsani, Daniel R Figueiredo, and Matthias Grossglauser. A bayesian method for matching two similar graphs without seeds. In *Communication, Control, and Computing (Allerton), 2013 51st Annual Allerton Conference on*, pages 1598–1607. IEEE, 2013.

[51] Ehsan Kazemi. Network alignment: Theory, algorithms, and applications. 2016.

[52] Pedram Pedarsani and Matthias Grossglauser. On the privacy of anonymized networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1235–1243. ACM, 2011.

[53] Ehsan Kazemi, Lyudmila Yartseva, and Matthias Grossglauser. When can two unlabeled networks be aligned under partial overlap? In *Communication, Control, and Computing (Allerton), 2015 53rd Annual Allerton Conference on*, pages 33–42. IEEE, 2015.

[54] Daniel Cullina and Negar Kiyavash. Improved achievability and converse bounds for erdos-renyi graph matching. In *ACM SIGMETRICS Performance Evaluation Review*, volume 44, pages 63–72. ACM, 2016.

[55] Shouling Ji, Weiqing Li, Neil Zhenqiang Gong, Prateek Mittal, and Raheem A Beyah. On your social network de-anonymizablity: Quantification and large scale evaluation with seed knowledge. In *NDSS*, 2015.

[56] Shouling Ji, Weiqing Li, Mudhakar Srivatsa, and Raheem Beyah. Structural data de-anonymization: Quantification, practice, and implications. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 1040–1053. ACM, 2014.

[57] Gábor György Gulyás, Benedek Simon, and Sándor Imre. An efficient and robust social network de-anonymization attack. In *Proceedings of the 2016 ACM on Workshop on Privacy in the Electronic Society*, pages 1–11. ACM, 2016.

[58] Lyudmila Yartseva and Matthias Grossglauser. On the performance of percolation graph matching. In *Proceedings of the first ACM conference on Online social networks*, pages 119–130. ACM, 2013.

[59] Shirin Nilizadeh, Apu Kapadia, and Yong-Yeol Ahn. Community-enhanced de-anonymization of online social networks. In *Proceedings of the 2014 acm sigsac conference on computer and communications security*, pages 537–548. ACM, 2014.

[60] Reza Shokri, George Theodorakopoulos, Carmela Troncoso, Jean-Pierre Hubaux, and Jean-Yves Le Boudec. Protecting location privacy: optimal strategy against localization attacks. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 617–627. ACM, 2012.

[61] Keith Henderson, Brian Gallagher, Lei Li, Leman Akoglu, Tina Eliassi-Rad, Hanghang Tong, and Christos Faloutsos. It's who you know: graph mining using recursive structural features. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 663–671. ACM, 2011.

[62] Michael Hay, Gerome Miklau, David Jensen, Philipp Weis, and Siddharth Srivastava. Anonymizing social networks. *Computer science department faculty publication series*, page 180, 2007.

[63] Antonio Criminisi, Jamie Shotton, Ender Konukoglu, et al. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends® in Computer Graphics and Vision*, 7(2–3):81–227, 2012.

[64] Peter J Haas. Data-stream sampling: basic techniques and results. In *Data Stream Management*, pages 13–44. Springer, 2016.

[65] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321–357, June 2002.

[66] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[67] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 20(8):832–844, 1998.

[68] Carla Fabiana, Michele Garetto, and Emilio Leonardi. De-anonymizing scale-free social networks by percolation graph matching. In *Computer Communications (INFOCOM), 2015 IEEE Conference on*, pages 1571–1579. IEEE, 2015.

[69] Naomi Altman and Martin Krzywinski. Points of significance: Association, correlation and causation, 2015.

[70] Galit Shmueli. To explain or to predict? *Statistical science*, pages 289–310, 2010.

[71] David R Brillinger. Some data analyses using mutual information. *Brazilian Journal of Probability and Statistics*, pages 163–182, 2004.

[72] P Bonissone, M Henrion, L Kanal, and J Lemmer. Equivalence and synthesis of causal models. In *Uncertainty in artificial intelligence*, volume 6, page 255, 1991.

[73] Adam kelleher. Causality. https://github.com/akelleh/causality, 2017.

[74] Jure Leskovec and Julian J Mcauley. Learning to discover social circles in ego networks. In *Advances in neural information processing systems*, pages 539–547, 2012.

[75] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):2, 2007.

[76] M. Fire, R. Puzis, and Y. Elovici. Link prediction in highly fractional data sets. *Handbook of Computational Approaches to Counterterrorism*, 2012.

[77] Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[78] Chiara Orsini, Marija M Dankulov, Pol Colomer-de Simón, Almerima Jamakovic, Priya Mahadevan, Amin Vahdat, Kevin E Bassler, Zoltán Toroczkai, Marián Boguñá, Guido Caldarelli, et al. Quantifying randomness in real networks. *Nature communications*, 6:8627, 2015.

[79] Priya Mahadevan, Dmitri Krioukov, Kevin Fall, and Amin Vahdat. Systematic topology analysis and generation using degree correlations. In *ACM SIGCOMM Computer Communication Review*, volume 36, pages 135–146. ACM, 2006.

[80] Minas Gjoka, Maciej Kurant, and Athina Markopoulou. 2.5 k-graphs: from sampling to generation. In *INFOCOM, 2013 Proceedings IEEE*, pages 1968–1976. IEEE, 2013.

[81] Paul W Holland and Samuel Leinhardt. An exponential family of probability distributions for directed graphs. *Journal of the american Statistical association*, 76(373):33–50, 1981.

[82] Stanley Wasserman and Philippa Pattison. Logit models and logistic regressions for social networks: I. an introduction to markov graphs andp. *Psychometrika*, 61(3):401–425, 1996.

[83] R Core Team. R: A language and environment for statistical computing. vienna, austria: R foundation for statistical computing; 2014, 2014.

[84] M Handcock, David R Hunter, Carter T Butts, S Goodreau, P Krivitsky, Skye Bender-deMoll, and Martina Morris. statnet: Software tools for the statistical analysis of network data. *The Statnet Project (http://www. statnet. org). R package version*, 2014.

[85] David R Hunter, Mark S Handcock, Carter T Butts, Steven M Goodreau, and Martina Morris. ergm: A package to fit, simulate and diagnose exponential-family models for networks. *Journal of statistical software*, 24(3):nihpa54860, 2008.

[86] Martina Morris, Mark S Handcock, and David R Hunter. Specification of exponential-family random graph models: terms and computational aspects. *Journal of statistical software*, 24(4):1548, 2008.

[87] Irene Sendiña-Nadal, Michael M Danziger, Z Wang, Shlomo Havlin, and Stefano Boccaletti. Assortativity and leadership emerge from anti-preferential attachment in heterogeneous networks. *Scientific reports*, 6:21297, 2016.

[88] Sameera Horawalavithana and Adriana Iamnitchi. On the privacy of dk-random graphs. *CoRR*, abs/1907.01695, 2019.

[89] Kourtellis N. Skvoretz J. Ripeanu M. Blackburn, J. and A. Iamnitchi. Cheating in online games: A social network perspective. *ACM Transactions on Internet Technology*, 13(3):9:1–9:25, 2014.

[90] Smith-Lovin L. McPherson, M. and J. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27:415–444, 2001.

[91] Neil Zhenqiang Gong, Ameet Talwalkar, Lester Mackey, Ling Huang, Eui Chul Richard Shin, Emil Stefanov, Elaine Runting Shi, and Dawn Song. Joint link prediction and attribute inference using a social-attribute network. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(2):27, 2014.

[92] Jianwei Qian, Xiang-Yang Li, Chunhong Zhang, and Linlin Chen. De-anonymizing social networks and inferring private attributes using knowledge graphs. In *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*, pages 1–9. IEEE, 2016.

[93] S. Ji, T. Wang, J. Chen, W. Li, P. Mittal, and R. Beyah. De-sag: On the de-anonymization of structure-attribute graph data. *IEEE Transactions on Dependable and Secure Computing*, PP(99):1–1, 2017.

[94] Luke K McDowell and David W Aha. Labels or attributes?: rethinking the neighbors for collective classification in sparsely-labeled networks. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 847–852. ACM, 2013.

[95] Lada A Adamic and Natalie Glance. The political blogosphere and the 2004 us election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, pages 36–43. ACM, 2005.

[96] Amanda L Traud, Peter J Mucha, and Mason A Porter. Social structure of facebook networks. *Physica A: Statistical Mechanics and its Applications*, 391(16):4165–4180, 2012.

[97] Lubos Takac and Michal Zabovsky. Data analysis in public social networks. In *International Scientific Conference and International Workshop Present Day Trends of Innovations*, number 6 in 1, 2012.

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

[99] John Skvoretz. Diversity, integration, and social ties: Attraction versus repulsion as drivers of intra- and intergroup relations. *American Journal of Sociology*, 119:486–517, 2013.

[100] WO Kermack and AG Mckendrick. A contribution to the mathematical theory of epidemics. *Proc Roy Soc*, 5, 2003.

[101] JT Hamrick, Farhang Rouhi, Arghya Mukherjee, Amir Feder, Neil Gandal, Tyler Moore, and Marie Vasek. The economics of cryptocurrency pump and dump schemes. 2018.

[102] Jiahua Xu and Benjamin Livshits. The anatomy of a cryptocurrency pump-and-dump scheme. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pages 1609–1625, 2019.

[103] Network of pro-maduro twitter accounts pushed anti-guaidó hashtags, Jan 2020.

[104] Stephan Hartmann. The world as a process. In *Modelling and simulation in the social sciences from the philosophy of science point of view*, pages 77–100. Springer, 1996.

[105] Joshua M Epstein. Agent-based computational models and generative social science. *Complexity*, 4(5):41–60, 1999.

[106] John Bollenbacher, Diogo Pacheco, Pik-Mai Hui, Yong-Yeol Ahn, Alessandro Flammini, and Filippo Menczer. On the challenges of predicting microscopic dynamics of online conversations. *Applied Network Science*, 6(1):1–21, 2021.

[107] Tarek Abdelzaher, Jiawei Han, Yifan Hao, Andong Jing, Dongxin Liu, Shengzhong Liu, Hoang Hai Nguyen, David M Nicol, Huajie Shao, Tianshi Wang, et al. Multiscale online media simulation with socialcube. *Computational and Mathematical Organization Theory*, pages 1–30, 2020.

[108] Hansika Hewamalage, Christoph Bergmeir, and Kasun Bandara. Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, 37(1):388–427, 2021.

[109] Anthony Hernandez, Kin Ng, and Adriana Iamnitchi. Using deep learning for temporal forecasting of user activity on social media: Challenges and limitations. In *Companion Proceedings of the Web Conference 2020*, pages 331–336, 2020.

[110] Sharad Goel, Duncan J Watts, and Daniel G Goldstein. The structure of online diffusion networks. In *Proceedings of the 13th ACM conference on electronic commerce*, pages 623–638, 2012.

[111] Manlio De Domenico and Eduardo G Altmann. Unraveling the origin of social bursts in collective attention. *Scientific reports*, 10(1):1–9, 2020.

[112] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *The International Conference on Learning Representations (ICLR)*, 2017.

[113] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *The 27th International Joint Conference on Artificial Intelligence*, 2017.

[114] Xiaofeng Gao, Zhenhao Cao, Sha Li, Bin Yao, Guihai Chen, and Shaojie Tang. Taxonomy and evaluation for microblog popularity prediction. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 13(2):1–40, 2019.

[115] Justin Cheng, Lada Adamic, P Alex Dow, Jon Michael Kleinberg, and Jure Leskovec. Can cascades be predicted? In *Proceedings of the 23rd international conference on World wide web*, pages 925–936. ACM, 2014.

[116] Bongwon Suh, Lichan Hong, Peter Pirolli, and Ed H Chi. Want to be retweeted? large scale analytics on factors impacting retweet in twitter network. In *2010 IEEE Second International Conference on Social Computing*, pages 177–184. IEEE, 2010.

[117] Nasir Naveed, Thomas Gottron, Jérôme Kunegis, and Arifah Che Alhadi. Bad news travel fast: A content-based analysis of interestingness on twitter. In *Proceedings of the 3rd international web science conference*, pages 1–7, 2011.

[118] Maximilian Jenders, Gjergji Kasneci, and Felix Naumann. Analyzing and predicting viral tweets. In *Proceedings of the 22nd international conference on world wide web*, pages 657–664, 2013.

[119] Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. Sentiment in twitter events. *Journal of the American Society for Information Science and Technology*, 62(2):406–418, 2011.

[120] Sasa Petrovic, Miles Osborne, and Victor Lavrenko. Rt to win! predicting message propagation in twitter. *ICWSM*, 11:586–589, 2011.

[121] Meng Jiang, Peng Cui, Rui Liu, Qiang Yang, Fei Wang, Wenwu Zhu, and Shiqiang Yang. Social contextual recommendation. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 45–54, 2012.

[122] Shuai Gao, Jun Ma, and Zhumin Chen. Effective and effortless features for popularity prediction in microblogging network. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 269–270, 2014.

[123] Gabor Szabo and Bernardo A Huberman. Predicting the popularity of online content. *Communications of the ACM*, 53(8):80–88, 2010.

[124] Benjamin Shulman, Amit Sharma, and Dan Cosley. Predictability of popularity: Gaps between prediction and understanding. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 10, 2016.

[125] Liangjie Hong, Ovidiu Dan, and Brian D Davison. Predicting popular messages in twitter. In *Proceedings of the 20th international conference companion on World wide web*, pages 57–58, 2011.

[126] Cheng Li, Jiaqi Ma, Xiaoxiao Guo, and Qiaozhu Mei. Deepcas: An end-to-end predictor of information cascades. In *Proceedings of the 26th International Conference on World Wide Web*, pages 577–586. International World Wide Web Conferences Steering Committee, 2017.

[127] Mohammad Raihanul Islam, Sathappan Muthiah, Bijaya Adhikari, B Aditya Prakash, and Naren Ramakrishnan. Deepdiffuse: Predicting the'who'and'when'in cascades. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1055–1060. IEEE, 2018.

[128] R. Krohn and T. Weninger. Modelling online comment threads from their start. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 820–829, 2019.

[129] Simon Bourigault, Sylvain Lamprier, and Patrick Gallinari. Representation learning for information diffusion through social networks: an embedded cascade model. In *Proceedings of the 9th ACM International Conference on Web Search and Data Mining*, pages 573–582. ACM, 2016.

[130] Victoria Zayats and Mari Ostendorf. Conversation modeling on reddit using a graph-structured lstm. *Transactions of the Association for Computational Linguistics*, 6:121–132, 2018.

[131] Pablo Aragón, Vicenç Gómez, David García, and Andreas Kaltenbrunner. Generative models of online discussion threads: state of the art and research challenges. *Journal of Internet Services and Applications*, 8(1):15, 2017.

[132] Alexey N. Medvedev, Renaud Lambiotte, and Jean-Charles Delvenne. The anatomy of reddit: An overview of academic research. In Fakhteh Ghanbarnejad, Rishiraj Saha Roy, Fariba Karimi, Jean-Charles Delvenne, and Bivas Mitra, editors, *Dynamics On and Of Complex Networks III*, pages 183–204, Cham, 2019. Springer International Publishing.

[133] Chen Ling, Guangmo Tong, and Mozi Chen. Nestpp: Modeling thread dynamics in online discussion forums. In *Proceedings of the 31st ACM Conference on Hypertext and Social Media*, pages 251–260, 2020.

[134] Chunyan Wang, Mao Ye, and Bernardo A Huberman. From user comments to on-line conversations. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 244–252. ACM, 2012.

[135] Pablo Aragón, Vicenç Gómez, and Andreaks Kaltenbrunner. To thread or not to thread: The impact of conversation threading on online discussion. In *11th International AAAI Conference on Web and Social Media*, 2017.

[136] Alexey N Medvedev, Jean-Charles Delvenne, and Renaud Lambiotte. Modelling structure and predicting dynamics of discussion threads in online boards. *Journal of Complex Networks*, 7(1):67–82, 2018.

[137] Ali Zarezade, Ali Khodadadi, Mehrdad Farajtabar, Hamid R. Rabiee, and Hongyuan Zha. Correlated cascades: Compete or cooperate. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 238–244, 2017.

[138] Isabel Valera and Manuel Gomez-Rodriguez. Modeling adoption and usage of competing products. In *Proceedings of the 2015 IEEE International Conference on Data Mining (ICDM)*, ICDM '15, pages 409–418, Washington, DC, USA, 2015. IEEE Computer Society.

[139] Siddharth Krishnan, Patrick Butler, Ravi Tandon, Jure Leskovec, and Naren Ramakrishnan. Seeing the forest for the trees: new approaches to forecasting cascades. In *Proceedings of the 8th ACM Conference on Web Science*, pages 249–258, 2016.

[140] Xinran He, Guojie Song, Wei Chen, and Qingye Jiang. Influence blocking maximization in social networks under the competitive linear threshold model. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, pages 463–474. SIAM, 2012.

[141] Wei Lu, Wei Chen, and Laks VS Lakshmanan. From competition to complementarity: comparative influence diffusion and maximization. *Proceedings of the VLDB Endowment*, 9(2):60–71, 2015.

[142] Yunpeng Xiao, Li Zhang, Qian Li, and Ling Liu. Mm-sis: Model for multiple information spreading in multiplex network. *Physica A: Statistical Mechanics and its Applications*, 513:135–146, 2019.

[143] Mohammad Raihanul Islam, Sathappan Muthiah, and Naren Ramakrishnan. Nactseer: Predicting user actions in social network using graph augmented neural network. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1793–1802, 2019.

[144] Jia Wang, Vincent W Zheng, Zemin Liu, and Kevin Chen-Chuan Chang. Topological recurrent neural network for diffusion prediction. In *Data Mining (ICDM), 2017 IEEE International Conference on*, pages 475–484. IEEE, 2017.

[145] Long Chen and Huifang Deng. Predicting user retweeting behavior in social networks with a novel ensemble learning approach. *IEEE Access*, 8:148250–148263, 2020.

[146] Seth A Myers and Jure Leskovec. Clash of the contagions: Cooperation and competition in information diffusion. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 539–548. IEEE, 2012.

[147] Lillian Weng, Alessandro Flammini, Alessandro Vespignani, and Fillipo Menczer. Competition among memes in a world with limited attention. *Scientific reports*, 2:335, 2012.

[148] Weiwei Liu, Zhi-Hong Deng, Xiuwen Gong, Frank Jiang, and Ivor Tsang. Effectively predicting whether and when a topic will become prevalent in a social network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[149] Hongzhi Yin, Bin Cui, Hua Lu, Yuxin Huang, and Junjie Yao. A unified model for stable and temporal topic detection from social media data. In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, pages 661–672. IEEE, 2013.

[150] Shoubin Kong, Qiaozhu Mei, Ling Feng, Fei Ye, and Zhe Zhao. Predicting bursts and popularity of hashtags in real-time. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 927–930, 2014.

[151] Pedro Saleiro and Carlos Soares. Learning from the news: Predicting entity popularity on twitter. In *International Symposium on Intelligent Data Analysis*, pages 171–182. Springer, 2016.

[152] Subhabrata Dutta, Sarah Masud, Soumen Chakrabarti, and Tanmoy Chakraborty. Deep exogenous and endogenous influence combination for social chatter intensity prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1999–2008, 2020.

[153] Prasha Shrestha, Suraj Maharjan, Dustin Arendt, and Svitlana Volkova. Learning from dynamic user interaction graphs to forecast diverse social behavior. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2033–2042, 2019.

[154] Luca Belli, Sofia Ira Ktena, Alykhan Tejani, Alexandre Lung-Yut-Fon, Frank Portman, Xiao Zhu, Yuanpu Xie, Akshay Gupta, Michael Bronstein, Amra Delić, et al. Privacy-aware recommender systems challenge on twitter's home timeline. *arXiv e-prints*, pages arXiv–2004, 2020.

[155] Aminu Da'u and Naomie Salim. Recommendation system based on deep learning methods: a systematic review and new directions. *Artificial Intelligence Review*, 53(4):2709–2748, 2020.

[156] Lesly Alejandra Gonzalez Camacho and Solange Nice Alves-Souza. Social network data to alleviate cold-start in recommender system: A systematic review. *Information Processing & Management*, 54(4):529–544, 2018.

[157] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 931–940, 2008.

[158] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 287–296, 2011.

[159] Xinxi Wang and Ye Wang. Improving content-based and hybrid music recommendation using deep learning. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 627–636, 2014.

[160] Hanjun Dai, Yichen Wang, Rakshit Trivedi, and Le Song. Deep coevolutionary network: Embedding user and item features for recommendation. *Knowledge Discovery and Data Mining (KDD)*, 2017.

[161] Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1269–1278, 2019.

[162] Vito Walter Anelli, Amra Delić, Gabriele Sottocornola, Jessie Smith, Nazareno Andrade, Luca Belli, Michael Bronstein, Akshay Gupta, Sofia Ira Ktena, Alexandre Lung-Yut-Fong, et al. Recsys 2020 challenge workshop: Engagement prediction on twitter's home timeline. In *Fourteenth ACM Conference on Recommender Systems*, pages 623–627, 2020.

[163] Wenzhe Shi and Luca Belli. What twitter learned from the recsys 2020 challenge, 2020.

[164] Benedikt Schifferer, Gilberto Titericz, Chris Deotte, Christof Henkel, Kazuki Onodera, Jiwei Liu, Bojan Tunguz, Even Oldridge, Gabriel De Souza Pereira Moreira, and Ahmet Erdem. Gpu accelerated feature engineering and training for recommender systems. In *Proceedings of the Recommender Systems Challenge 2020*, pages 16–23. 2020.

[165] Ajay Kumar, Shashank Sheshar Singh, Kuldeep Singh, and Bhaskar Biswas. Link prediction techniques, applications, and performance: A survey. *Physica A: Statistical Mechanics and its Applications*, 553:124289, 2020.

[166] Jimeng Sun and Jie Tang. A survey of models and algorithms for social influence analysis. In *Social network data analytics*, pages 177–214. Springer, 2011.

[167] William L Hamilton. Graph representation learning. *Synthesis Lectures on Artifical Intelligence and Machine Learning*, 14(3):1–159, 2020.

[168] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *The International Conference on Learning Representations (ICLR)*, 2017.

[169] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *The International Conference on Learning Representations (ICLR)*, 2018.

[170] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 1025–1035, Red Hook, NY, USA, 2017. Curran Associates Inc.

[171] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *The International Conference on Learning Representations (ICLR)*, 2019.

[172] Boris Weisfeiler and Andrei Leman. The reduction of a graph to canonical form and the algebra which appears therein. *NTI, Series*, 2(9):12–16, 1968.

[173] Qian Huang, Horace He, Abhay Singh, Ser-Nam Lim, and Austin R Benson. Combining label propagation and simple models out-performs graph neural networks. *arXiv preprint arXiv:2010.13993*, 2020.

[174] Defense Advanced Research Projects Agency DARPA. Computational simulation of online social behavior (socialsim). https://www.darpa.mil/program/computational-simulation-of-online-social-behavior, 2021.

[175] Ivan Garibay, Toktam A Oghaz, Niloofar Yousefi, Ece C Mutlu, Madeline Schiappa, Steven Scheinert, Georgios C Anagnostopoulos, Christina Bouwens, Stephen M Fiore, Alexander Mantzaris, et al. Deep agent: Studying the dynamics of information spread and evolution in social networks. *arXiv preprint arXiv:2003.11611*, 2020.

[176] Irving John Good. Weight of evidence, corroboration, explanatory power, information and the utility of experiments. *Journal of the Royal Statistical Society: Series B (Methodological)*, 22(2):319–331, 1960.

[177] Kevin A Gluck and Richard W Pew. *Modeling human behavior with integrated cognitive architectures: Comparison, evaluation, and validation*. Psychology Press, 2006.

[178] Sameera Horawalavithana, John Skvoretz, and Adriana Iamnitchi. Cascade-lstm: Predicting information cascades using deep neural networks. *arXiv preprint arXiv:2004.12373*, 2020.

[179] Linyun Yu, Peng Cui, Fei Wang, Chaoming Song, and Shiqiang Yang. From micro to macro: Uncovering and predicting information cascading process with behavioral dynamics. In Charu C. Aggarwal, Zhi-Hua Zhou, Alexander Tuzhilin, Hui Xiong, and Xindong Wu, editors, *2015 IEEE International Conference on Data Mining, ICDM 2015, Atlantic City, NJ, USA, November 14-17, 2015*, pages 559–568. IEEE Computer Society, 2015.

[180] Sharad Goel, Ashton Anderson, Jake Hofman, and Duncan J Watts. The structural virality of online diffusion. *Management Science*, 62(1):180–196, 2015.

[181] Yunfei Lu, Linyun Yu, Tianyang Zhang, Chengxi Zang, Peng Cui, Chaoming Song, and Wenwu Zhu. Collective human behavior in cascading system: Discovery, modeling and applications. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 297–306. IEEE, 2018.

[182] Justin Cheng, Lada A Adamic, Jon M Kleinberg, and Jure Leskovec. Do cascades recur? In *Proceedings of the 25th International Conference on World Wide Web*, pages 671–681. International World Wide Web Conferences Steering Committee, 2016.

[183] Kenneth De Jong. Machine learning. chapter Genetic-algorithm-based Learning, pages 611–638. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

[184] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[185] François Chollet et al. Keras. https://keras.io, 2015.

[186] Maria Glenski, Emily Saldanha, and Svitlana Volkova. Characterizing speed and scale of cryptocurrency discussion spread on reddit. In *The World Wide Web Conference*, pages 560–570, 2019.

[187] Wikipedia. Bitcoin scalability problem. https://en.wikipedia.org/wiki/Bitcoin_scalability_problem/, 2019.

[188] Clayton J Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *8th international AAAI conference on weblogs and social media*, 2014.

[189] Alberto Lumbreras. *Automatic role detection in online forums*. PhD thesis, 2016.

[190] Ravi Kumar, Mohammad Mahdian, and Mary McGlohon. Dynamics of conversations. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 553–562. ACM, 2010.

[191] Vicenç Gómez, Hilbert J Kappen, Nelly Litvak, and Andreas Kaltenbrunner. A likelihood-based framework for the analysis of discussion threads. *World Wide Web*, 16(5-6):645–675, 2013.

[192] Seth A Myers, Chenguang Zhu, and Jure Leskovec. Information diffusion and external influence in networks. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 33–41, 2012.

[193] Renee DiResta, Kris Shaffer, Becky Ruppel, David Sullivan, Robert Matney, Ryan Fox, Jonathan Albright, and Ben Johnson. *The Tactics & Tropes of the Internet Research Agency*. New Knowledge, 2018.

[194] Sameera Horawalavithana. Mcas. https://github.com/SamTube405/mcas, 2021.

[195] Manuel Gomez-Rodriguez, Le Song, Hadi Daneshmand, and Bernhard Schölkopf. Estimating diffusion networks: Recovery conditions, sample complexity & soft-thresholding algorithm. *The Journal of Machine Learning Research*, 17(1):3092–3120, 2016.

[196] Kate Starbird and Leysia Palen. " voluntweeters" self-organizing by digital volunteers in times of crisis. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1071–1080, 2011.

[197] Firoj Alam, Ferda Ofli, and Muhammad Imran. Crisismmd: Multimodal twitter datasets from natural disasters. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 12, 2018.

[198] Nauman Saeed, Suku Sinnappan, and Stefanie Markham. The tale of two cultures: differences in technology acceptance in twitter usage. In *ACIS 2012: Location, location, location: Proceedings of the 23rd Australasian Conference on Information Systems 2012*, pages 1–9. ACIS, 2012.

[199] Richard Rogers. Deplatforming: Following extreme internet celebrities to telegram and alternative social media. *European Journal of Communication*, 35(3):213–229, 2020.

[200] Tarleton Gillespie. *Custodians of the Internet: Platforms, content moderation, and the hidden decisions that shape social media*. Yale University Press, 2018.

[201] Abir De, Sourangshu Bhattacharya, and Niloy Ganguly. Demarcating endogenous and exogenous opinion diffusion process on social networks. In *Proceedings of the 2018 World Wide Web Conference*, pages 549–558, 2018.

[202] Zeynep Zengin Alp and Şule Gündüz Öğüdücü. Identifying topical influencers on twitter based on user behavior and network topology. *Knowledge-Based Systems*, 141:211–221, 2018.

[203] Soroush Vosoughi, Mostafa 'Neo' Mohsenvand, and Deb Roy. Rumor gauge: Predicting the veracity of rumors on twitter. *ACM transactions on knowledge discovery from data (TKDD)*, 11(4):1–36, 2017.

[204] Filipe N Ribeiro, Matheus Araújo, Pollyanna Gonçalves, Marcos André Gonçalves, and Fabrício Benevenuto. Sentibench-a benchmark comparison of state-of-the-practice sentiment analysis methods. *EPJ Data Science*, 5(1):1–29, 2016.

[205] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 2019.

[206] Kalev Leetaru and Philip A Schrodt. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA annual convention*, volume 2, pages 1–49. Citeseer, 2013.

[207] Elizaveta Levina and Peter Bickel. The earth mover's distance is the mallows distance: Some insights from statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 251–256. IEEE, 2001.

[208] Lydia Manikonda, Ghazaleh Beigi, Huan Liu, and Subbarao Kambhampati. Twitter for sparking a movement, reddit for sharing the moment:# metoo through the lens of social media. *International Conference on Social Computing, Behavioral-Cultural Modeling Prediction and Behavior Representation in Modeling and Simulation*, 2018.

[209] Shalini Priya, Ryan Sequeira, Joydeep Chandra, and Sourav Kumar Dandapat. Where should one get news updates: Twitter or reddit. *Online Social Networks and Media*, 9:17 – 29, 2019.

[210] Sameera Horawalavithana, Kin Wai Ng, and Adriana Iamnitchi. Twitter is the megaphone of cross-platform messaging on the white helmets. In *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*, pages 235–244. Springer, 2020.

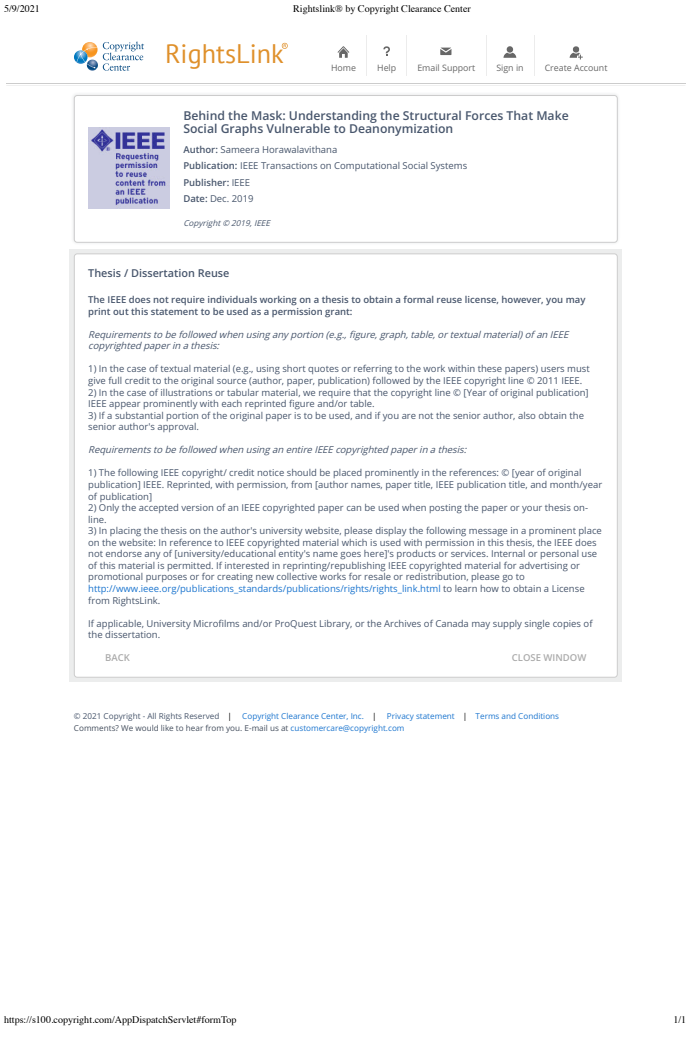
[211] Sameera Horawalavithana, Abhishek Bhattacharjee, Renhao Liu, Nazim Choudhury, Lawrence O. Hall, and Adriana Iamnitchi. Mentions of security vulnerabilities on reddit, twitter and github. In *IEEE/WIC/ACM International Conference on Web Intelligence*, pages 200–207, 2019.

[212] Kate Starbird, Ahmer Arif, Tom Wilson, Katherine Van Koevering, Katya Yefimova, and Daniel Scarnecchia. Ecosystem or echo-system? exploring content sharing across alternative media domains. In *AAAI ICWSM*, 2018.

[213] Xianwen Kuang. Central state vs. local levels of government: Understanding news media censorship in china. *Chinese Political Science Review*, 3(2):154–171, 2018.

Appendix A: Copyright Permissions

The permission as shown below is for the reuse of content as appeared in the portions of abstract, Chapters 1, and 2.



5/9/2021 Rightslink® by Copyright Clearance Center

Copyright Clearance Center RightsLink®

Home | Help | Email Support | Sign in | Create Account

IEEE Requesting permission to reuse content from an IEEE publication

Behind the Mask: Understanding the Structural Forces That Make Social Graphs Vulnerable to Deanonymization

Author: Sameera Horawalavithana
Publication: IEEE Transactions on Computational Social Systems
Publisher: IEEE
Date: Dec. 2019

Copyright © 2019, IEEE

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW

https://s100.copyright.com/AppDispatchServlet#formTop 1/1

The permission as shown below is for the reuse of content as appeared in the portions of abstract, Chapters 1, and 3.

Author of this Springer Nature content	yes
Title	Diversity, Homophily and the Risk of Node Re-identification in Labeled Social Graphs
Institution name	University of South Florida
Expected presentation date	Jun 2021
Requestor Location	Mr. Yasanka Horawalavithana [redacted] TAMPA, FL 33617 United States Attn: Mr. Yasanka Horawalavithana
Total	0.00 USD

Terms and Conditions

Springer Nature Customer Service Centre GmbH
Terms and Conditions

This agreement sets out the terms and conditions of the licence (the **Licence**) between you and **Springer Nature Customer Service Centre GmbH** (the **Licensor**). By clicking 'accept' and completing the transaction for the material (**Licensed Material**), you also confirm your acceptance of these terms and conditions.

1. Grant of License

1. 1. The Licensor grants you a personal, non-exclusive, non-transferable, world-wide licence to reproduce the Licensed Material for the purpose specified in your order only. Licences are granted for the specific use requested in the order and for no other use, subject to the conditions below.

1. 2. The Licensor warrants that it has, to the best of its knowledge, the rights to license reuse of the Licensed Material. However, you should ensure that the material you are requesting is original to the Licensor and does not carry the copyright of another entity (as credited in the published version).

1. 3. If the credit line on any part of the material you have requested indicates that it was reprinted or adapted with permission from another source, then you should also seek permission from that source to reuse the material.

2. Scope of Licence

2. 1. You may only use the Licensed Content in the manner and to the extent permitted by these Ts&Cs and any applicable laws.

2. 2. A separate licence may be required for any additional use of the Licensed Material, e.g. where a licence has been purchased for print only use, separate permission must be obtained for electronic re-use. Similarly, a licence is only valid in the language selected and does not apply for editions in other languages unless additional translation rights have been granted separately in the licence. Any content owned by third parties are expressly excluded from the licence.

2. 3. Similarly, rights for additional components such as custom editions and derivatives require additional permission and may be subject to an additional fee. Please apply to Journalpermissions@springernature.com/bookpermissions@springernature.com for these rights.

2. 4. Where permission has been granted **free of charge** for material in print, permission may also be granted for any electronic version of that work, provided that the material is incidental to your work as a whole and that the electronic version is essentially equivalent to, or substitutes for, the print version.

2. 5. An alternative scope of licence may apply to signatories of the STM Permissions Guidelines, as amended from time to time.

3. Duration of Licence

3. 1. A licence for is valid from the date of purchase ('Licence Date') at the end of the relevant period in the below table:

Scope of Licence	Duration of Licence
Post on a website	12 months
Presentations	12 months
Books and journals	Lifetime of the edition in the language purchased

4. Acknowledgement

4. 1. The Licensor's permission must be acknowledged next to the Licenced Material in print. In electronic form, this acknowledgement must be visible at the same time as the figures/tables/illustrations or abstract, and must be hyperlinked to the journal/book's homepage. Our required acknowledgement format is in the Appendix below.

5. Restrictions on use

5. 1. Use of the Licensed Material may be permitted for incidental promotional use and minor editing privileges e.g. minor adaptations of single figures, changes of format, colour and/or style where the adaptation is credited as set out in Appendix 1 below. Any other changes including but not limited to, cropping, adapting, omitting material that affect the meaning, intention or moral rights of the author are strictly prohibited.

5. 2. You must not use any Licensed Material as part of any design or trademark.

5. 3. Licensed Material may be used in Open Access Publications (OAP) before publication by Springer Nature, but any Licensed Material must be removed from OAP sites prior to final publication.

6. Ownership of Rights

6. 1. Licensed Material remains the property of either Licensor or the relevant third party and any rights not explicitly granted herein are expressly reserved.

7. Warranty

IN NO EVENT SHALL LICENSOR BE LIABLE TO YOU OR ANY OTHER PARTY OR ANY OTHER PERSON OR FOR ANY SPECIAL, CONSEQUENTIAL, INCIDENTAL OR INDIRECT DAMAGES, HOWEVER CAUSED, ARISING OUT OF OR IN CONNECTION WITH THE DOWNLOADING, VIEWING OR USE OF THE MATERIALS REGARDLESS OF THE FORM OF ACTION, WHETHER FOR BREACH OF CONTRACT, BREACH OF WARRANTY, TORT, NEGLIGENCE, INFRINGEMENT OR OTHERWISE (INCLUDING, WITHOUT LIMITATION, DAMAGES BASED ON LOSS OF PROFITS, DATA, FILES, USE, BUSINESS OPPORTUNITY OR CLAIMS OF THIRD PARTIES), AND
WHETHER OR NOT THE PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION SHALL APPLY NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY PROVIDED HEREIN.

8. Limitations

8. 1. ***BOOKS ONLY:*** Where **'reuse in a dissertation/thesis'** has been selected the following terms apply: Print rights of the final author's accepted manuscript (for clarity, NOT the published version) for up to 100 copies, electronic rights for use only on a personal website or institutional repository as defined by the Sherpa guideline (www.sherpa.ac.uk/romeo/).

8. 2. For content reuse requests that qualify for permission under the STM Permissions Guidelines, which may be updated from time to time, the STM Permissions Guidelines supersede the terms and conditions contained in this licence.

9. Termination and Cancellation

9. 1. Licences will expire after the period shown in Clause 3 (above).

9. 2. Licensee reserves the right to terminate the Licence in the event that payment is not received in full or if there has been a breach of this agreement by you.

Appendix 1 — Acknowledgements:

For Journal Content:
Reprinted by permission from [**the Licensor**]: [**Journal Publisher** (e.g. Nature/Springer/Palgrave)] [**JOURNAL NAME**] [**REFERENCE CITATION** (Article name, Author(s) Name), [**COPYRIGHT**] (year of publication)

For **Advance Online Publication papers:**
Reprinted by permission from [**the Licensor**]: [**Journal Publisher** (e.g.

Nature/Springer/Palgrave)] [**JOURNAL NAME**] [**REFERENCE CITATION** (Article name, Author(s) Name), [**COPYRIGHT**] (year of publication), advance online publication, day month year (doi: 10.1038/sj.[JOURNAL ACRONYM].)

For Adaptations/Translations:
Adapted/Translated by permission from [**the Licensor**]: [**Journal Publisher** (e.g. Nature/Springer/Palgrave)] [**JOURNAL NAME**] [**REFERENCE CITATION** (Article name, Author(s) Name), [**COPYRIGHT**] (year of publication)

Note: For any republication from the British Journal of Cancer, the following credit line style applies:

Reprinted/adapted/translated by permission from [**the Licensor**]: on behalf of Cancer Research UK: : [**Journal Publisher** (e.g. Nature/Springer/Palgrave)] [**JOURNAL NAME**] [**REFERENCE CITATION** (Article name, Author(s) Name), [**COPYRIGHT]** (year of publication)

For **Advance Online Publication** papers:
Reprinted by permission from The [**the Licensor**]: on behalf of Cancer Research UK: [**Journal Publisher** (e.g. Nature/Springer/Palgrave)] [**JOURNAL NAME**] [**REFERENCE CITATION** (Article name, Author(s) Name), [**COPYRIGHT**] (year of publication), advance online publication, day month year (doi: 10.1038/sj.[JOURNAL ACRONYM])

For Book content:
Reprinted/adapted by permission from [**the Licensor**]: [**Book Publisher** (e.g. Palgrave Macmillan, Springer etc) [**Book Title**] by [**Book author**(s)] [**COPYRIGHT**] (year of publication)

Other Conditions:

Version 1.3

Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.

SPRINGER NATURE

The risk of node re-identification in labeled social graphs

Author: Sameera Horawalavithana et al

Publication: Applied Network Science

Publisher: Springer Nature

Date: Jun 13, 2019

The permission as shown below is for the reuse of content as appeared in the portions of abstract, Chapters 1, 4 and 6.

ACM Author Gateway

Author Resources

Home > **Author Resources** > **Author Rights & Responsibilities**

ACM Author Rights

ACM exists to support the needs of the computing community. For over sixty years ACM has developed publications and publication policies to maximize the visibility, impact, and reach of the research it publishes to a global community of researchers, educators, students, and practitioners. ACM has achieved its high impact, high quality, widely-read portfolio of publications with:

- Affordably priced publications
- Liberal Author rights policies
- Wide-spread, perpetual access to ACM publications via a leading-edge technology platform
- Sustainability of the good work of ACM that benefits the profession

Choose

Authors have the option to choose the level of rights management they prefer. ACM offers three different options for authors to manage the publication rights to their work.

- Authors who want ACM to manage the rights and permissions associated with their work, which includes defending against improper use by third parties, can use ACM's traditional copyright transfer agreement.
- Authors who prefer to retain copyright of their work can sign an exclusive licensing agreement, which gives ACM the right but not the obligation to defend the work against improper use by third parties.
- Authors who wish to retain all rights to their work can choose ACM's author-pays option, which allows for perpetual open access through the ACM Digital Library. Authors choosing the author-pays option can give ACM non-exclusive permission to publish, sign ACM's exclusive licensing agreement or sign ACM's traditional copyright transfer agreement. Those choosing to grant ACM a non-exclusive permission to publish may also choose to display a Creative Commons License on their works.

Post

Otherwise known as "Self-Archiving" or "Posting Rights", all ACM published authors of magazine articles, journal articles, and conference papers retain the right to post the pre-submitted (also known as "pre-prints"), submitted, accepted, and peer-reviewed versions of their work in any and all of the following sites:

- Author's Homepage
- Author's Institutional Repository
- Any Repository legally mandated by the agency or funder funding the research on which the work is based
- Any Non-Commercial Repository or Aggregation that does not duplicate ACM tables of contents. Non-Commercial Repositories are defined as Repositories owned by non-profit organizations that do not charge a fee to access deposited articles and that do not sell advertising or otherwise profit from serving scholarly articles.

For the avoidance of doubt, an example of a site ACM authors may post all versions of their work to, with the exception of the final published "Version of Record", is ArXiv. ACM does request authors, who post to ArXiv or other permitted sites, to also post the published version's Digital Object Identifier (DOI) alongside the pre-published version on these sites, so that easy access may be facilitated to the published "Version of Record" upon publication in the ACM Digital Library.

Examples of sites ACM authors may not post their work to are ResearchGate, Academia.edu, Mendeley, or Sci-Hub, as these sites are all either commercial or in some instances utilize predatory practices that violate copyright, which negatively impacts both ACM and ACM authors.

Distribute

Authors can post an Author-Izer link enabling free downloads of the Definitive Version of the work permanently maintained in the ACM Digital Library.

- On the Author's own Home Page or
- In the Author's Institutional Repository.

Reuse

Authors can reuse any portion of their own work in a new work of their own (and no fee is expected) as long as a citation and DOI pointer to the Version of Record in the ACM Digital Library are included.

- Contributing complete papers to any edited collection of reprints for which the author is notthe editor, requires permission and usually a republication fee.
- Authors can include partial or complete papers of their own (and no fee is expected) in a dissertation as long as citations and DOI pointers to the Versions of Record in the ACM Digital Library are included. Authors can use any portion of their own work in presentations and in the classroom (and no fee is expected).
- Commercially produced course-packs that are sold to students require permission and possibly a fee.

Create

ACM's copyright and publishing license include the right to make Derivative Works or new versions. For example, translations are "Derivative Works." By copyright or license, ACM may have its publications translated. However, ACM

Authors continue to hold perpetual rights to revise their own works without seeking permission from ACM.

Minor Revisions and Updates to works already published in the ACM Digital Library are welcomed with the approval of the appropriate Editor-in-Chief or Program Chair.

- If the revision is minor, i.e., less than 25% of new substantive material, then the work should still have ACM's publishing notice, DOI pointer to the Definitive Version, and be labeled a "Minor Revision of"
- If the revision is major, i.e., 25% or more of new substantive material, then ACM considers this a new work in which the author retains full copyright ownership (despite ACM's copyright or license in the original published article) and the author need only cite the work from which this new one is derived.

Retain

Authors retain all perpetual rights laid out in the ACM Author Rights and Publishing Policy, including, but not limited to:

- Sole ownership and control of third-party permissions to use for artistic images intended for exploitation in other contexts
- All patent and moral rights
- Ownership and control of third-party permissions to use of software published by ACM