User Vulnerability and its Reduction on a Social Networking Site

PRITAM GUNDECHA, Arizona State University
GEOFFREY BARBIER[1], Air Force Research Laboratory
JILIANG TANG, Arizona State University
HUAN LIU, Arizona State University

Privacy and security are major concerns for many users of social media. When users share information (e.g., data and photos) with friends, they can make their friends vulnerable to security and privacy breaches with dire consequences. With the continuous expansion of a user's social network, privacy settings alone are often inadequate to protect user's profile. In this research, we aim to address some critical issues related to privacy protection: (1) How can we measure and assess individual user's vulnerability ? (2) With the diversity of one's social network friends, how can one figure out an effective approach to maintaining balance between vulnerability and social utility? In this work, first we present a novel way to define vulnerable friends from an individual user's perspective. User vulnerability is dependent on whether or not the user's friends' privacy settings protect the friend and the individual's network of friends (which includes the user). We show that it is feasible to measure and assess user vulnerability, and reduce one's vulnerability without changing the structure of a social networking site. The approach is to unfriend one's most vulnerable friends. However, when such a vulnerable friend is also socially important, unfriending him would significantly reduce one's own social status. We formulate this novel problem as vulnerability minimization with social utility constraints. We formally define the optimization problem, and provide an approximation algorithm with a proven bound. Finally, we conduct a large-scale evaluation of new framework using a Facebook dataset. We resort to experiments and observe how much vulnerability an individual user can decrease by unfriending a vulnerable friend. We compare performance of different unfriending strategies and discuss the security risk of new friend request. Additionally, by employing different forms of social utility, we confirm that balance between user vulnerability and social utility can be practically achieved.

Categories and Subject Descriptors: H.2.7 [**Information Systems**]: Security, integrity, and protection; J.4 [**Social and Behavioral Sciences**]: Sociology

General Terms: Security, Experimentation

Additional Key Words and Phrases: Vulnerability, Social network, Privacy

ACM Reference Format:
Gundecha P., Barbier G., Tang J. and Liu H. 2014. User Vulnerability and its Reduction on a Social Networking Site. ACM Trans. Embedd. Comput. Syst. 9, 4, Article 39 (March 2010), 26 pages.
DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

[1]The majority of this work was conducted when Geoffrey Barbier was affiliated with the Computer Science Department at Arizona Sate University

This work is supported by grants of ARO (025071), ONR (N000141010091, N000141410095) and AFOSR (FA95500810132). This work was also funded, in part, by OSD-T&E (Office of Secretary Defense-Test and Evaluation), DefenseWide/PE0601120D8Z National Defense Education Program (NDEP)/BA-1, Basic Research; SMART Program Office, www.asee.org/fellowships/smart, Grant Number N00244-09-1-0081.
Author's address: P. Gundecha, J. Tang and H. Liu, Computer Science Department, Arizona State University, Geoffrey Barbier, Air Force Research Laboratory (AFRL)

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

1. INTRODUCTION

Social media has gained popularity in recent years, becoming an integral part of our lives. It enables users to expand their ways of communications in reporting news, expressing sentiments, exchanging opinions, and making online friends [Gundecha and Liu 2012; Zafarani et al. 2014] This trend should continue for the foreseeable future. Social media has tremendous potential to help or hurt us [Morozov 2011]. With the presence of adversaries, the convenience of and low barriers of access to social networking sites bring about new challenges. On a social networking site, an individual user can share a large amount of personal and sometimes sensitive information with friends through channels such as the user's profile, frequent status updates, messages, and status replies. Depending on individual choice, the user profile can reveal personal information to friends such as gender, birth date, relationship status, e-mail address, phone number, home address, and even political or religious affiliations [Gundecha et al. 2013]. This puts an implicit responsibility on a user's friends to keep shared information private and honor the implicit, and sometimes explicit, trust placed on friends by the user.

A *social networking site* [Boyd and Ellison 2008] is a web-based service that allows web users to publish a public or semi-public profile within a bounded system, divulge a network of friends with whom they share a connection, and explore other users' profiles and friend networks. Social networking sites have reshaped business models [Vaynerchuk 2009], provided platform for communities to grow [Tang et al. 2008; Tang and Liu 2009], stimulated viral marketing [Subramani and Rajagopalan 2003; Leskovec et al. 2007], provided trend analysis and sales prediction [Sterne 2010], and can be a grass-roots information source [Goolsby 2010].

Though the fundamental goal of social networking sites is to enable users to be more social, user privacy and security issues cannot be ignored. On one hand, most social networking sites provide limited means to protect users' privacy and security. On the other hand, social networking users are often unaware that they could pose a threat to their friends. In this paper, we show that it is feasible to measure a user's vulnerability based on three factors: (1) user privacy settings can reveal personal information; (2) a user's action on a social networking site can expose their friends' personal information; and (3) friends' action on a social networking site can reveal user's personal information. Based on these three factors, we later show that how user's vulnerability can be measured and assessed. This vulnerability measure enables us to quantify users vulnerability, and identify their vulnerable friends. As we draw parallels between users and their friends, we are interested in finding an effective mechanism that could make users less vulnerable. Unfriending with vulnerable friends reduces users vulnerability. This mechanism has been validated using extensive experiments on users and their friends on Facebook.

Sociologists, psychologists and economists [Parsons 1968; Veblen and Banta 2007; Becker 1974; Brock and Durlauf 2001] have been researching the impact of social interactions on the social utility value of a user and the society. Although unfriending vulnerable friends can reduce vulnerability, this simple strategy can limit social interaction opportunities among users. Besides limiting interaction opportunities, unfriending socially important or valuable friends can backfire and reduce one's social status as well. Social importance can be measured in terms of social utility [Loewenstein et al. 1989]. One such utility is the nodal degree of a friend's. Refer to Figure 1: if A is the most vulnerable but also most popular among U's friends, could U unfriend his other vulnerable friends instead of A in order to reduce vulnerability? Herewith, the additional new challenge is how to maintain low vulnerability and high social utility for a social networking user. The work in this paper addresses the challenge by developing

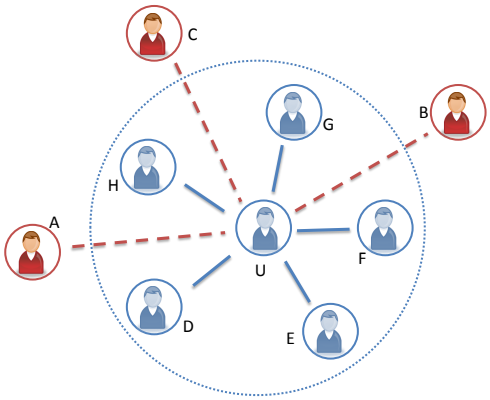


Fig. 1: Which one vulnerable friend to unfriend from (A, B, C) for user U?

novel solutions to the following questions without suggesting any structural change to a social networking site.

(1) How can we measure and assess user's vulnerability? Is there an effective mechanism to make users less vulnerable?
(2) What is the social cost of vulnerability reduction mechanism on user's social utility ? How can we achieve balance between user's vulnerability and social utility?

Rest of the paper is organized as follows. In Section 2, we review the literature highlighting the novelty of this effort. In Section 3, we study the collectible statistics from a social networking site and present a quantifiable measure to evaluate users vulnerability and define the problem of identifying vulnerable friends. We propose a methodology and measures for evaluating whether or not a user is vulnerable and how to adjust a user's network to best deal with threats presented by vulnerable friends. In Section 4, we present a constrained vulnerability minimization problem. To this end, we formulate two novel optimization problems of vulnerability minimization. We also discuss the hardness of the problem and provide approximation guarantees to efficient algorithms. In Section 5, we conduct an empirical study to evaluate methods that can be manipulated to make users less vulnerable, compare the performance of an optimal algorithm with that of intuitive heuristic methods, and discuss the approach that can be used to assess the impact of new friends to a user's network. We also evaluate methods that make users less vulnerable while retaining acceptable level of social utility values of vulnerable friends. Finally, we conclude the paper with possible future research directions in Section 6.

2. RELATED WORK

Work discussed in this paper is about *identifying vulnerable friends of a user at one social networking site* and differs from most previous efforts of securing users privacy on a social networking site. To the best of our knowledge, we are first to present the quantifiable measure to evaluate users vulnerability.

Previous research has shown that users' private behavior can be analyzed based on cues, such as user's web site browsing logs [Hu et al. 2007; Goel et al. 2012], contents of personal web sites [Marcus et al. 2006], music collections [Rentfrow and Gosling 2003], and social media profiles from Twitter [Quercia et al. 2011; Golbeck et al. 2011]. Kosinski et.al. [Kosinski et al. 2013] show that wide variety of people's highly sensitive personal attributes can be automatically and accurately inferred using their Facebook

Likes alone. Personal attributes include sexual orientation, ethnicity, religious and political views, personality traits, intelligence, happiness, use of addictive substances, parental separation, age and gender. This paper focus on publicly available user profiles and their network information on Facebook.

A range of privacy settings as well as control mechanisms are available on all social networking sites. These settings help users protect their personal information to some extent, but are often not straightforward to reset and how to do so is not always communicated effectively. Previous research [Gross and Acquisti 2005] shows us that only a few users change the default privacy preferences on Facebook. In some cases, user profiles are completely public, making information available and providing a communication mechanism to anyone who wants to access it. It is no secret that when a profile is made public, malicious users including stalkers, spammers, and hackers can use sensitive information for their personal gain. Sometimes malevolent users can even cause physical or emotional distress to other users [Rosenblum 2007].

On most social networking sites, privacy-related efforts have been concentrated on protecting individual profile attributes only. However, users are often left vulnerable to various attacks, as there is not enough protection for them from their friends. In this work, we show that privacy settings alone may not be sufficient to protect privacy. Social media users can also face privacy breach if their friends abuse their trust. Research on deanonymization of social network data shows that the deanonymization process benefits from publicly available profile information and social activities of users on a social networking site. Research [Narayanan and Shmatikov 2008; 2009] demonstrates how users' privacy can be weakened if an attacker knows the presence of connections among users. Research [Wondracek et al. 2010] also presents a successful scheme that exploits only the group membership information of users to breach privacy. This underlines the fact that users alone cannot completely protect their profile even if they mark their profile as private.

Liu and Maes [Liu and Maes 2005] point out a lack of privacy awareness and find a large number of social network profiles in which people describe themselves with a rich vocabulary in terms of their passions and interests. Krishnamurthy and Wills [Krishnamurthy and Wills 2010] discuss the problem of leakage of personally identifiable information and how it can be misused by third parties [Narayanan and Shmatikov 2009]. There is research that suggests some fundamental changes necessary to social networking sites to achieve user privacy. Squicciarini et al. [Squicciarini et al. 2009] introduce a novel collective privacy mechanism for better managing shared content between the users. Previous work [Fang and LeFevre 2010] focuses on helping users to express simple privacy settings, but do not consider additional problems such as attribute inference [Zheleva and Getoor 2009] or shared data ownership [Squicciarini et al. 2009]. Researchers [Zheleva and Getoor 2009] show how an adversary can exploit an online social network with a mixture of public and private user profiles to predict the private attributes of users. There is a framework [Baden et al. 2009] where users dictate who may access their information based on public-private encryption-decryption algorithms. Although the proposed framework addresses privacy concerns, it comes with the cost of increased response time. Our work does not suggest any fundamental changes to social networking sites.

Recently, there have been a few research efforts focused on the different factors in unfriending [Sibona and Walczak 2011] decisions. In this work, we propose different unfriending[2] strategies to protect user privacy. This is relatively new area of research and to the best of our knowledge, we are the first one to employ the unfriending mechanism to reduce user vulnerability. Using traditional computer network, researchers propose

[2]http://www.nytimes.com/2010/10/24/fashion/24Studied.html

methods to identify nodes [Tong et al. 2010] and edges [Tong et al. 2012] in a large graph such that network can be robust against a dissemination. We also highlight the consequences of reducing user vulnerability, based on unfriending, without considering the social utility loss to a user. The social utility of a user can be defined by different measures of social network analysis. In the past, a spectrum preserving graph randomization method [Ying and Wu 2008] shows some network properties can be preserved while protecting edge anonymity. In this work, we use three basic measures: nodal degree or simply degree, tie strength, and number of common friends. A friend with high degree means she is a popular one; when two have a strong tie [Gilbert and Karahalios 2009] or have a large number of common friends, they are very close friends. In other words, we can employ these measures help determine the consequence of unfriending on a user. Unfriending a vulnerable friend with high degree could limit the user's potential to make more friends, which makes user more reclusive and defeats the purpose of social networking. Generally, family members, best friends, and girlfriend or boyfriend are examples of strong ties. Though unfriending them could reduce one's vulnerability, it would not be desirable. When the user and his friend share a large number of friends, removing the friend could affect structural balance [Easley and Kleinberg 2010] of the social network and the user's clustering coefficient [Watts and Strogatz 1998]. Thus, it is essential to consider the social utility of vulnerable friends before unfriending them in order to reduce vulnerability.

3. DEFINING USER VULNERABILITY AND HOW TO REDUCE IT

Every user on a social networking site can choose to reveal their personal information using a range of attributes. Figure 1 shows an illustrative example where a user U has eight friends (A, B, C, D, E, F, G, H). Based on their preferences, friends assumed to be revealing different attributes from the available lists of personal attributes. In this section, we first propose a measure to quantify user U's vulnerability, and then introduce a model of vulnerability reduction. We first divide the user's personal attributes into two sets, *individual and community attributes*. Individual attributes (I-attributes) characterize individual user information, including personal information such as gender, birth date, phone number, home address, group memberships, etc. Community attributes (C-attributes) characterize information about friends of a user, including friends that are traceable from a user's profile (i.e., user's friend list), tagged pictures, wall interactions, etc. These attributes are always accessible to friends but may not to the other users. A user's vulnerability depends on the visibility and exposure of a user's profile through not only attributes settings but also his friends.

3.1. Defining User Vulnerability

Oftentimes users on a social networking site are unaware that they could pose a threat to their friends due to their vulnerability. In this paper, we show that it is feasible to measure a user's vulnerability based on three factors: (1) user's privacy settings that can reveal personal information; (2) a user's action on a social networking site that can expose their friends' personal information; and (3) friends' action on a social networking site that can reveal user's personal information. Based on these factors, we formally present one of the earliest models for vulnerability reduction.

Definition 3.1 (*I-index*). I-index estimates how much risk to privacy a user can incur by allowing individual attributes to be accessible or visible to other users. A user who ignores or is unaware of privacy settings is a threat to himself. I-index is defined as a function of individual attributes (I-attributes). I-index of user u is given by

$$I_u = f(A_u), \tag{1}$$

where $I_u \in [0,1]$, $A_u = \{a_{u,i} | \forall i, a_{u,i} \in \{0,1\}\}$ is I-attribute set for user u, and $a_{u,i}$ is a status of a i-th I-attribute for a user u. $a_{u,i} = 1$ indicates user u has enabled i-th I-attribute to be visible to everyone otherwise non-visible (may be sensitive for a user). Note a user attribute visible to only friends is marked as disabled.

Table I shows statistics of commonly found I-attributes on Facebook (Refer to Section 5.1 for details on Facebook dataset consists of 2,056,646 users). The last column in the table lists the percentage of people who enable the particular attribute to be visible. For example, 7,430 (0.36%) Facebook users enabled their mobile phone numbers to be visible. We define the sensitivity (weight), of an attribute as a percentage of non-visibility. Hence, the sensitivity of a mobile phone number according to our Facebook dataset is 99.64. This means that users do not usually disclose their mobile phone number to other users. Users that do disclose phone numbers have a propensity to vulnerability because they disclose more sensitive information in their profiles.

Attributes	User Count	Percentage (%)
Total users	2,056,646	
I-attributes:		
Current City	620,401	30.17
Hometown	727,674	35.38
Gender	1,681,673	81.77
Birthday	67,834	3.30
Relationship status	539,612	26.24
Siblings	244,658	11.90
Education and work	516,848	25.13
Like and interests	1,369,080	66.57
Email	27,103	1.32
Mobile number	7,430	0.36
Website	128,776	6.26
Home address	7,580	0.37
Political Views	24,438	1.19
Religious Views	33,036	1.61
Children	86,609	4.21
Networks	284,482	13.83
Parents	73,887	3.49
Bio	199,070	9.68
Interested in	383,724	18.66
Looking for	449,498	21.86
Music	941,340	45.77
Books	281,346	13.68
Movies	574,243	27.92
Television	684,843	33.30
Activities	385,417	18.74
Interests	308,229	14.99
C-attributes:		
Friends trace (link)	1,481,472	72.03

Table I: Attributes statistics on the Facebook

We used normalized weighted average to estimate I-index. I-index for each profile user u is given by,

$$I_u = f(A_u) = \frac{\sum_{i=1}^{n} w_i * a_{u,i}}{\sum_{i=1}^{n} w_i}, \tag{2}$$

where w_i is the sensitivity (weight) of an i-th I-attribute, n is the total number of I-attributes available via a social networking site profile and $a_{u,i} = 1$ if i-th I-attribute is visible otherwise the attribute is not visible (i.e., sensitive to user u). $I_u \in [0, 1]$. $I_u = 1$ indicates user u has marked all I-attributes to be visible. On the other hand, $I_u = 0$ indicates user u has marked all I-attributes to be non-visible.

From the viewpoint of optimization, it is common to use linear sum as an objective function or constraints to reduce the overall complexity of finding the optimal solution (e.g. Linear Programming). Inspired from this, the paper proposes a linear sum (weighted average) function of individual attributes and their sensitivity weight (percentage of non-visibility) to compute the I-index of a user (Equation (2)). The proposed linear sum function is a simple, and captures the intuition that vulnerability of a user increases with more visibility of some attributes. For example, Table I shows that the profile revealing "religious (1.61%)" and "political (1.19%)" affiliation values should be more vulnerable in comparison with the profile revealing "Gender (81.77%)" and "Relationship status (26.24%)" values.

Definition 3.2 (*C-index*). C-index estimates how much threat a user can pose to their friends by making community attributes accessible or visible to other users. Users who ignore and are unaware of privacy settings of community attributes can create risk to the entire community of friends. C-index is defined as a function of community attributes (C-attributes). C-index for a user u is given by

$$C_u = g(B_u), \tag{3}$$

where $C_u \in [0, 1]$, $B_u = \{b_{u,i} | \forall i, b_{u,i} \in \mathbb{Z}^+\}$ is C-attributes set for user u, $b_{u,i}$ indicates the number of friends affected when a corresponding C-attribute is manifested, and $\mathbb{Z}^+$ is the set of positive integers. We ignore attributes marked as non-visible. Our Facebook dataset has only one C-attribute (see Table I) which suggests how many friends are traceable (via a friend relationship) from an individual user. 1,481,472 (72.03%) Facebook users in our dataset allowed friends to trace to other users. Thus, a large portion of users are *either not careful or not aware of the privacy concerns of their friends*.

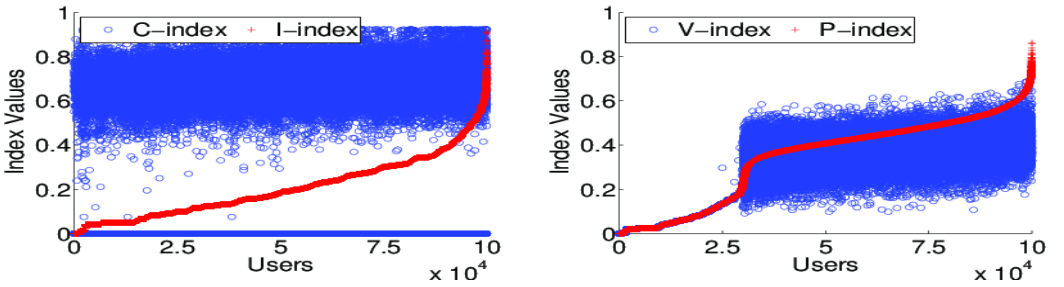
A vulnerable user u can pose threat to his friends and the amount of the threat increases with the number of friends that are put at risk. However, the rate of the increment decreases as more friends are put at risk. To appropriately represent this threat change, we choose a convex, non-decreasing log function to estimate the threat for each user based on the number friends placed at risk by each C-attribute. Hence, C-index for a user u is calculated as

$$C_u = g(B_u) = \frac{\sum_{i=1}^{m} \log(b_{u,i})}{4 * m}, \tag{4}$$

where m is the total number of C-attributes possible on a social networking site, and constant 4 is chosen because $C_u \in [0, 1]$ and none of the Facebook users in our dataset has more than 10^4 friends. $C_u > 0$ indicates user u has allowed everyone to trace friends through their own profile. On the other hand, $C_u = 0$ indicates that all the friends (except one) of a user u are non-traceable through a profile.

Fig. 2a shows I-index and C-index for randomly chosen 100K Facebook users. Note that users are sorted in ascending order of their I-indexes which gives curve-like impres-

sion on plotting I-index. The X-axis and Y-axis indicate users and their corresponding I and C-index values, respectively. Fig. 2a demonstrates that for the majority of users, the C-index value is greater than the corresponding I-index value. This highlights the one finding of this paper that a large portion of users are either not careful or not aware of the privacy concerns of their friends.



(a) I-index (red) and C-index (blue) for each User.

(b) P-index (red) and V-index (blue) for each User.

Fig. 2: Relationship among index values for each User.

Definition 3.3 (*P-index*). It estimates how public (visible) or private (non-visible) a user is on a social networking site. It shows how much an individual user aims to protect himself as well as his friends. P-index is defined as a function of I-index and C-index. P-index of user u is given by

$$P_u = j(I_u, C_u), \tag{5}$$

where $P_u \in [0, 1]$.

We choose a simple, weighted average function to calculate P-index for each Facebook user in our dataset.

$$P_u = \alpha * I_u + (1 - \alpha) * C_u = \alpha * f(A_u) + (1 - \alpha) * g(B_u), \tag{6}$$

where $\alpha \in [0, 1]$. Substituting Eq(6) with Eq(2) and Eq(4), we get

$$P_u = \alpha * \frac{\sum_{i=1}^{n} w_i * a_{u,i}}{\sum_{i=1}^{n} w_i} + (1 - \alpha) * \frac{\sum_{i=1}^{m} \log(b_{u,i})}{4 * m} \tag{7}$$

Different users may have different priorities about friends and may have different perspectives about vulnerability. Tunable parameter α can be set to address the needs of different users. For example, one may choose $\alpha < 0.5$ to deemphasize the individual attributes' visibility; or one may choose $\alpha > 0.5$ to emphasize the individual attributes' visibility. For our experiments, we set $\alpha = 0.5$ to put equal weights to individual and community attributes.

P-index address the first two factors of the user vulnerability estimation discussed above, i.e., user's privacy settings (I-index) and actions to expose friends (C-index). Thus, we follow a commonly used optimization formulation as in linear program as we did for I-index. Mathematically any function which can combine the I-index and C-index and ranges between [0,1] can be used. In this paper, the above simple weighted average function works well and it can also be easily discerned in applying these indices for various needs.

Definition 3.4 (*V-index*). V-index estimates how vulnerable a user is on a social networking site. Thus far, we have provided three indexes, I-index, C-index, and P-index, for a user based on the visibility of I-attributes and C-attributes. Vulnerability of a user depends on privacy settings of self, friends, their friends, and so on. Intuitively, as the distance between a user and other users on a social networking site increases, the marginal risk of vulnerability decreases the further away a user is from a vulnerable user. Hence, we only consider a user and friends in estimating the vulnerability of a user. V-index of a user depends on the P-indexes of friends and him. V-index of user u is defined as

$$V_u = h(P_{F_u \cup \{u\}}), \tag{8}$$

where F_u is the set of friends of user u, $P_{F_u \cup \{u\}} = \{P_i | i \in F_u \cup \{u\}\}$, and $V_u \in [0, 1]$.

We rewrite the above notation without loss of generality as

$$V_u = h(F_u \cup \{u\}) \tag{9}$$

Figure 2b shows the P-index and V-index for 100K randomly chosen users (the same users chosen for Figure 2a). Note that users are sorted in ascending order of their P-indexes which gives curve-like impression on plotting P-index. The X-axis and Y-axis indicate users and their index values respectively. A simple, weighted average function is used to plot V-index for each user,

$$V_u = \frac{P_u + \sum_{i \in F_u} P_i}{|F_u| + 1} \tag{10}$$

Figure 2b demonstrates that for the majority of users, the V-index value is greater than the corresponding P-index value. This highlights the need for focusing on friends for reducing the V-index value.

The design of V-index function has a direct impact on the complexity of solving the problem of identifying 'k' vulnerable friends (described in detail next). We show later that this problem can be solved in polynomial time, if the V-index is computed as shown in Equation (10).

3.2. Reducing User Vulnerability

Next, we will provide the mechanism based on unfriending to reduce user vulnerability.

Definition 3.5 (*A vulnerable friend*). A user's vulnerable friend is defined as a friend whose unfriending will lower the V-index score of a user. The V-index of a user u upon removing the vulnerable friend v is given by

$$V'_u = h(F_u \cup \{u\} \setminus \{v\}) \tag{11}$$

By the definition of a vulnerable friend, $V'_u < V_u$.

The definition of a vulnerable friend can be generalized to k-vulnerable friends.

Definition 3.6 (k-*vulnerable friends*). k-vulnerable friends of a user are k friends whose unfriending will lower the V-index score of a user. The V-index of user, u, upon removing k vulnerable friends $\{v_1, \cdots, v_k\}$ is given by

$$V'_u = h(F_u \cup \{u\} \setminus \{v_1, \cdots, v_k\}), \tag{12}$$

By the definition of k-vulnerable friends, $V'_u < V_u$.

Based on Definitions 3.5 and 3.6, a user's friends can be divided into two sets: (1) an initial set of vulnerable friends $D_{u,0}$, who are responsible for increasing the V-index

value of user u, and (2) a set of non-vulnerable friends $F_u \setminus D_{u,0}$, who are responsible for decreasing the V-index value of user u. Hence, Eq(9) can be rewritten as

$$V_u = h(\{F_u \setminus D_{u,0}\} \cup \{u\} \cup D_{u,0}) \tag{13}$$

In order to minimize the user vulnerability[3] function $h(\cdot)$, we have to unfriend vulnerable friends $D_{u,0}$. The user vulnerability minimization problem seeks, for a parameter k from user u, to find a new set of k vulnerable friends $D_u \subseteq D_{u,0}$ to unfriend. The minimum new V-index for user u is achieved after unfriending the selected vulnerable friends D_u, where $|D_u| \leq k$. The new V-index of user, u, upon removing selected set of vulnerable friends $D_u \subseteq D_{u,0}$, is given by

$$V'_u = h(\{F_u \setminus D_{u,0}\} \cup \{u\} \cup \{D_{u,0} \setminus D_u\}) \tag{14}$$

By the definition of k-vulnerable friends, $V'_u < V_u$.

This problem of minimizing the vulnerability of user u is equivalently stated as finding the set of at most k vulnerable friends $D_u \subseteq D_{u,0}$ to unfriend, who are responsible for maximizing the vulnerability of user u.

Let $\sigma(D_u)$ be the estimate how vulnerable user u is due to vulnerable friends D_u. Thus, we maximize function $\sigma : 2^{|D_{u,0}|} \rightarrow R^*$, where $\mathbb{R}^*$ is the set of non-negative real numbers. Note that, $h(D_u)$ is not the same as $\sigma(D_u)$, even though function $\sigma(\cdot)$ depends on function $h(\cdot)$. For example, if $h(\cdot)$ is a simple average function of P-indexes of user and friends, then function $\sigma(\cdot)$ estimates the total P-index value of all the vulnerable friends. In other words, function $\sigma(D_u)$ estimates the vulnerability induced by the vulnerable friends D_u on user u. The Vulnerability Maximization with Cardinality constraint problem (VMC) is formulated as follows

VMC($D_{u,0}$, $\vec{P}$, k, $\mathcal{V}_u$) - *Instance:* The finite set of initial vulnerable friends $D_{u,0} \subseteq F_u$ of user u, P-index $P_i \in [0,1]\ \forall i \in D_{u,0}$, a vector $\vec{P} = (P_1, \cdots, P_{|D_{u,0}|})$, and constant $k \in \mathbb{Z}^*$ and $\mathcal{V}_u \in \mathbb{R}^*$. *Question:* Is there a subset $D_u \subseteq D_{u,0}$ such that $|D_u| \leq k$ and $\sigma(D_u) \geq \mathcal{V}_u$?

The above problem can be solved in polynomial time, if function $\sigma(\cdot)$ is a linear function of P-index values of vulnerable friends. The linear function $\sigma(\cdot)$ can be represented as

$$\sigma(D_{u,0}) = \sum_{i=1}^{|D_{u,0}|} \lambda_i * P_i, \quad \forall i \in D_{u,0}, \lambda_i \in \mathbb{R}^* \tag{15}$$

Since vulnerable friends make a user profile less secure, λ_i cannot be a negative real number. For Eq(15), the most vulnerable friend $d \in D_{u,0}$ is given by,

$$d = \max_{v \in D_{u,0}} \sigma(D_{u,0}) = \max_i \{\lambda_i * P_i | \forall i \in D_{u,0}\} \tag{16}$$

If we repetitively identify the most vulnerable friend d, using Eq(16), for k (or n, if $n < k$) times and remove d from $D_{u,0}$ for every run, we can get k-vulnerable friends to maximally reduce user vulnerability.

We do not assume that $\sigma(\cdot)$ is linear. Later, we will discuss how to solve **VMC** problem, when $\sigma(\cdot)$ is non-linear.

So far we aim to reduce the vulnerability of a user without considering its social impact. If the vulnerable user selected to unfriend is also socially valuable, then it could lead to a serious social problem. For example, a user may not want to unfriend his girlfriend, though vulnerable. Next we investigate this problem of user vulnerability reduction with social utility constraints.

[3]A user vulnerability can also be minimized by (1) disabling visibility of sensitive user's attributes (such as phone number, email address, home address, etc.) and not exposing friends to others; and by (2) requesting (or negotiating with) the vulnerable friends to lower their vulnerability index.

4. VULNERABILITY REDUCTION WITH SOCIAL UTILITY

An essential function of social networking sites is to help users to be social. Although unfriending vulnerable friends from a user's social network sometimes can reduce vulnerability, this strategy could sometimes significantly limit social interaction among users.

Vulnerable friends	P-index	Degree	Tie Strength	#Common Friends
A	0.9	100	0.8	30
B	0.7	200	0.5	50
C	0.5	300	0.3	10

Table II: Unfriending one vulnerable friend from (A, B, C) to minimize user U's vulnerability while retaining socially valuable friends. High P-index suggests high vulnerability. Social utility can be materialized in various forms. In this work, we use three common measures: one's nodal degree, tie strength, and number of common friends.

The social utility can be defined by different measures of social network analysis. In this work, we use three basic measures: nodal degree or simply degree, tie strength, and number of common friends. A friend with high degree means she is a popular one; when two friends have a strong tie [Gilbert and Karahalios 2009] or have a large number of common friends, they are very close friends. In other words, we can employ these measures help determine the consequence of unfriending on a user. Unfriending a vulnerable friend with high degree could limit the user's potential to make more friends, which makes user more reclusive and defeats the purpose of social networking. Generally, family members, best friends, and girl or boy friend are examples of strong ties. Though unfriending them could reduce one's vulnerability, it would not be desirable. When the user and his friend share a large number of friends, removing the friend could affect structural balance [Easley and Kleinberg 2010] of the social network and the user's clustering coefficient [Watts and Strogatz 1998]. Thus, it is essential to consider the social utility of vulnerable friends before unfriending them in order to reduce vulnerability.

There exist many social utility measures that can be categorized into two types [Backstrom et al. 2006]: (1) social utility measures related to the vulnerable friends, and (2) social utility measures related to the relationship between user and each of his vulnerable friends. Table III lists some social utility measures available for a user u to consider while selecting the vulnerable friends $D_u \subseteq D_{u,0}$ to unfriend. Figure 1 shows an illustrative example where a user U has three vulnerable friends (A, B, C). Table II lists users P-indexes (indicating user visibility) and their social utility measures including degree, tie strength and number of common friends. If we do not consider the social utility measures of vulnerable friends, the most vulnerable friend A should be removed to minimize user U's vulnerability. If we consider tie strength, user A is the most valuable for user U. Similarly, friends B and C are also valuable because B shares the most number of common friends and C has the highest degree. Under these different circumstances, we now study which friend U should unfriend in order to reduce vulnerability with the constraint of retaining social utility.

Next we define the optimization problem for reducing a user vulnerability while retaining socially valuable friends.

Definition 4.1 (*Social Utility Loss*). It estimates how much a user loses on a social networking site after unfriending friends. Social utility loss of user u depends on social

Social utility measures related to the initial set of vulnerable friends $D_{u,0}$ of a user u.	Degree (number of friends) of a vulnerable friend. Local clustering coefficient (density) of a vulnerable friend. Number of closed triads of a vulnerable friend. Number of open triads of a vulnerable friend. Number of posts (social activity) by a vulnerable friend. Number of responses (social activity) by a vulnerable friend. Popularity (may be based on replies on posts) of a vulnerable friend. Influence index of a vulnerable friend.
Social utility measures related to the relationship between user u and each vulnerable friend in $D_{u,0}$	Number of common friends between user and vulnerable friend. Tie strength between user and vulnerable friend. Number of uncommon friends between user and vulnerable friend. Number of responses (social activity) by a user u on a vulnerable friend's post. Number of responses (social activity) by a vulnerable friend on a user's post. Homophily (similarity) index between user and a vulnerable friend.

Table III: Types of Social Utility Measures

utility values of u's friends. For a given social utility measure, the social utility loss for a user u after unfriending with a given set of friends $A \subseteq F_u$ is given by

$$L_u = \zeta(S_A), \tag{17}$$

where $S_A = \{S_i | i \in A \subseteq F_u, S_i \in \mathbb{R}^*\}$, S_i represents the social utility measure of user i, and $L_u \in \mathbb{R}^*$.

Assuming $\zeta(\cdot)$ to be a simple additive function, we have

$$L_u = \sum_{i \in A \subseteq F_u} S_i \tag{18}$$

Similar to the **VMC** problem formulation, the problem of minimizing the vulnerability of user u with social utility constraint can be stated as finding the set of at most k vulnerable friends $D_u \subseteq D_{u,0}$ to unfriend, who are responsible for maximizing the user vulnerability, and minimizing user u's social utility loss. We can formally state the problem of Vulnerability Maximization with minimum social Utility loss and Cardinality constraint (**VMUC**) as

VMUC($D_{u,0}$, $\vec{P}$, $\vec{S}$, k, $\mathcal{V}_u$, $\mathcal{L}_u$): - *Instance:* Given a finite set of initial vulnerable friends $D_{u,0} \subseteq F_u$ of a user u, P-index $P_i \in [0,1], \forall i \in D_{u,0}$, a vector $\vec{P} = (P_1, \cdots, P_{|D_{u,0}|})$, social utility measure $S_i \in R^* \ \forall i \in D_{u,0}$, a vector $\vec{S} = (S_1, \cdots, S_{|D_{u,0}|})$, and constants $k \in \mathbb{Z}^*$, $\mathcal{V}_u \in \mathbb{R}^*$, and $\mathcal{L}_u \in \mathbb{R}^*$. - ***Question:*** Find a subset $D_u \subseteq D_{u,0}$ such that $|D_u| \leq k$, $\sigma(D_u) \geq \mathcal{V}_u$, and $\sum_{i \in D_u} S_i \leq \mathcal{L}_u$.

We now focus on solving the **VMUC**($D_{u,0}$**,** $\vec{P}$**,** $\vec{S}$**,** k**,** $\mathcal{V}_u$**,** $\mathcal{L}_u$) problem. First, we prove the function $\sigma(\cdot)$ is non-negative, non-decreasing and submodular.

THEOREM 4.2 (MONOTONICITY). *The function $\sigma : D_{u,0} \rightarrow \mathbb{R}^*$ is monotonically non-decreasing. i.e, $\sigma(A) \leq \sigma(A \cup \{v\})$, where $A \subseteq D_{u,0}$ and $v \in D_{u,0}$.*

Proof. As discussed above, for a user u, each friend can be classified into vulnerable or non-vulnerable friend. From Definition 3.5, the V-index of a user u decreases upon removing the vulnerable friend v. Therefore, $h(S \cup A) \leq h(S \cup A \cup \{v\})$, where S and A are sets of non-vulnerable (including user u) and vulnerable friends, respectively. This means that for user u, the vulnerability of set $A \cup \{v\}$ is more than that of set A. Hence, $\sigma(A) \leq \sigma(A \cup \{v\})$. □

THEOREM 4.3 (SUBMODULARITY). *If the function $h(\cdot)$ is submodular in terms of vulnerable friends, the function $\sigma(\cdot)$ is submodular, i.e., $\sigma(A \cup \{v\}) - \sigma(A) \geq \sigma(B \cup \{v\}) - \sigma(B)$, where $A \subseteq B \subseteq D_{u,0}$, and $v \in D_{u,0}$.*

Proof. If the function $h(\cdot)$ is submodular in terms of vulnerable friends, the marginal gain in vulnerability of user u, by adding a vulnerable friend v to an initial vulnerable set A, is at least as high as the marginal gain, by adding the same vulnerable node v to an initial vulnerable superset B, i.e., $h(S \cup A \cup \{v\}) - h(S \cup A) \geq h(S \cup B \cup \{v\}) - h(S \cup B)$, where S is the set of non-vulnerable friends (which includes user u), A and B are sets of vulnerable friends, and $A \subseteq B$. This means that the new vulnerable friend v causes more increase when added to a set A than to a superset B. Thus, $\sigma(\cdot)$ is submodular. □

Let us examine the assumption that the function $h(\cdot)$ is submodular in terms of vulnerable friends. Assume that user v is user u's vulnerable friend. If user u has 25 vulnerable friends as opposed to 50, where 25 vulnerable friends are a subset of 50, then based on the assumption about the function $h(\cdot)$, user v is more vulnerable for user u when u has fewer vulnerable friends than when u has more. In other words, the vulnerability of user v can be mitigated due to the presence of more u's vulnerable friends.

Based on Theorems 4.2 and 4.3, the **VMC** problem is tantamount to the maximization of non-negative, non-decreasing, submodular function with cardinality constraint. A hill climbing algorithm can solve this problem with provable constant approximation [Nemhauser et al. 1978]. First start with an empty output set D_u; add one element from an initial set of vulnerable friends $D_{u,0}$ to the output set that provides the largest marginal increase in the function value; repeat the previous step until all the elements from an initial set of vulnerable friends $D_{u,0}$ are processed or the maximum cardinality bound k is reached. According to [Nemhauser et al. 1978], this greedy algorithm gives a $(1 - 1/e)-$approximation for maximization of $\sigma(\cdot)$ function with a given cardinality constraint.

Similarly, with Theorems 4.2 and 4.3, the **VMUC** problem is equivalent to the maximization of non-negative, non-decreasing, submodular function with knapsack like constraints. The greedy algorithm presented in [Sviridenko 2004] can be applied to solve the **VMUC** problem with submodular objective function constrained by cardinality and social utility. The proposed algorithm also gives $(1 - 1/e)$-approximation guarantee.

The VMUC problem remains NP-hard [Garey and Johnson 1979] even when the objective function $\sigma(\cdot)$ is linear, constrained by social utility and cardinality. It can be reduced to a single dimensional knapsack problem. The following scaling and rounding algorithm is a fully polynomial time approximation scheme [Vazirani 2001] (FPTAS) for the VMUC problem with a linear objective function with knapsack like constraints: for each vulnerable user $i \in D_{u,0}$, define new P-index $P'_i = \lfloor \frac{P_i}{K} \rfloor$, where $K = \frac{\epsilon * P}{n}$, $n = |D_{u,0}|$, $P = \sum_{i \in D_{u,0}} P_i$ and a given error parameter $\epsilon > 0$; with the new P-index, using a

dynamic programming algorithm similar to the single dimensional knapsack problem, find the most vulnerable set $D_u \subseteq D_{u,0}$ and $|D_u| \leq k$.

For applying the scaling rounding algorithm, P-index values of all vulnerable friends need to be integers. The P-index value of each user $i \in D_{u,0}$ is a non negative real number in the closed range 0 to 1. Since our problem is a discrete optimization problem, we can simply convert these P-index values into integers by shifting the decimal points equally to the right. For experiments, on the **VMUC** problem with linear objective, we multiply each P-index value by 1000 and then take the floor of the resulting value as new integer value for P-index. Errors caused due to the scaling and rounding are negligible.

Solutions presented for **VMC** and **VMUC** problems, with corresponding assumptions, are summarized in the Table IV. The **VMUC** problem for non-linear social utility gain constraints remains an open problem.

		VMC	VMUC Problem
Objective function $\sigma(\cdot)$	Linear	1	FPTAS
	Submodular	$(1-1/e)$	$(1-1/e)$

Table IV: Best known approximation schemes and bounds for the VMC and VMUC problems. The objective function $\sigma(\cdot)$ is submodular provided that the function $h(\cdot)$ is also submodular in terms of vulnerable friends.

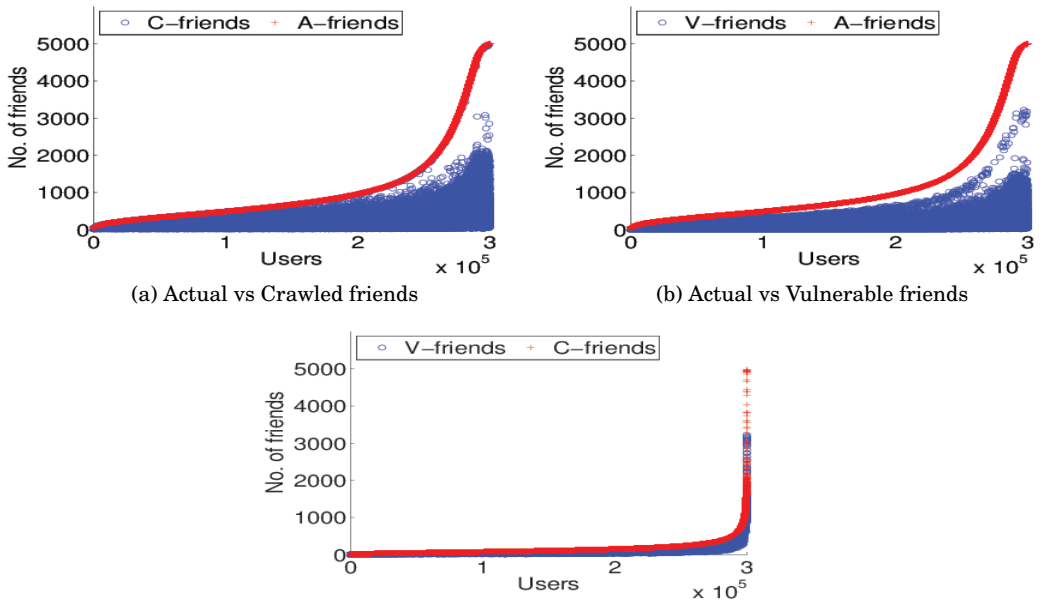
5. RESULTS

The proposed methods are demonstrated in practice through experiments using a dataset derived from a real social networking site. The proposed experiments address the challenge of vulnerability reduction with and without social utility constraints. With an approach for identifying vulnerable friends, we set out to investigate the following issues:

— How effective are the measures in reducing vulnerability of users? What is an effective way of reducing one's vulnerability? How does it compare random unfriending in reducing vulnerability of users?
— Do the indexes address the dynamics of social networks? We study the impact of a new friend request and its effect on vulnerability of a user.
— How effective are the unfriending algorithms in recommending at most k (≥ 0) vulnerable friends to minimize user vulnerability while maintaining an acceptable level[4] of social utility loss?
— Does the user vulnerability reduction change significantly for different social utility measures?
— At most, how many vulnerable friends should a user unfriend to achieve a desired vulnerability reduction while maintaining an acceptable level of social utility loss?

Next we discuss the dataset used for experiments, use the proposed index estimation methods in an empirical study in an attempt to address these issues, report preliminary results, and suggest new lines of research in finding vulnerable users.

[4]In this work, we set the acceptable level of social utility loss less than or equal to 10%



(a) Actual vs Crawled friends

(b) Actual vs Vulnerable friends

(c) Crawled vs Vulnerable friends

Fig. 3: Facebook dataset fact

5.1. Social Network Data

According to Quantcast[5], over 145 million unique users in United States visit Facebook within a month. This puts Facebook among top 3 websites based on the number of people in the United States who visit each site within a month. Facebook users spend over 700 billion minutes per month. The statistics suggest that Facebook[6] users provide a rich set of personal information through their profile, and social activities. Thus, we use a Facebook dataset for evaluating the proposed work.

The Facebook dataset[7] is created by crawling Facebook user profiles. Crawling is performed in breadth-first search manner starting from randomly selected users(roots). The dataset contains publicly available profiles as well as network information for more than two million users. We design two major tasks: vulnerability reduction without social utility constraints, and vulnerability reduction with social utility constraints. For the first task, we do not filter any Facebook user profiles from the dataset. However, for the second task we remove the Facebook users who do not share their friends' information from the dataset, since unfriending is not possible on such users. Table V shows the statistics of randomly selected 300K users from the dataset used for the second task.

For a given user, we may not obtain the information of all friends due to their privacy settings. Friends for which we obtain the information are referred as crawled friends. Figure 3 shows further facts about the randomly selected 300K users. X-axis and Y-axis indicate users and their number of friends, respectively. For simplicity, before plotting Figures 3a and 3b, we sort all users in the ascending order based on the

[5]http://www.quantcast.com/facebook.com, a media sharing and web analytics service company.
[6]http://newsroom.fb.com/content/default.aspx?NewsAreaId=22
[7]We use the same dataset as in [Gundecha et al. 2011]

Facebook dataset	Count
Avg. actual friends per user	1056
Avg. crawled friends per user	154
Avg. vulnerable friends per user	98
Max. actual friends per user	5000
Min. actual friends per user	11
Max. crawled friends per user	4971
Min. crawled friends per user	11
Max. vulnerable friends per user	3222
Min. vulnerable friends per user	0

Table V: Statistics for randomly selected 300K Facebook users

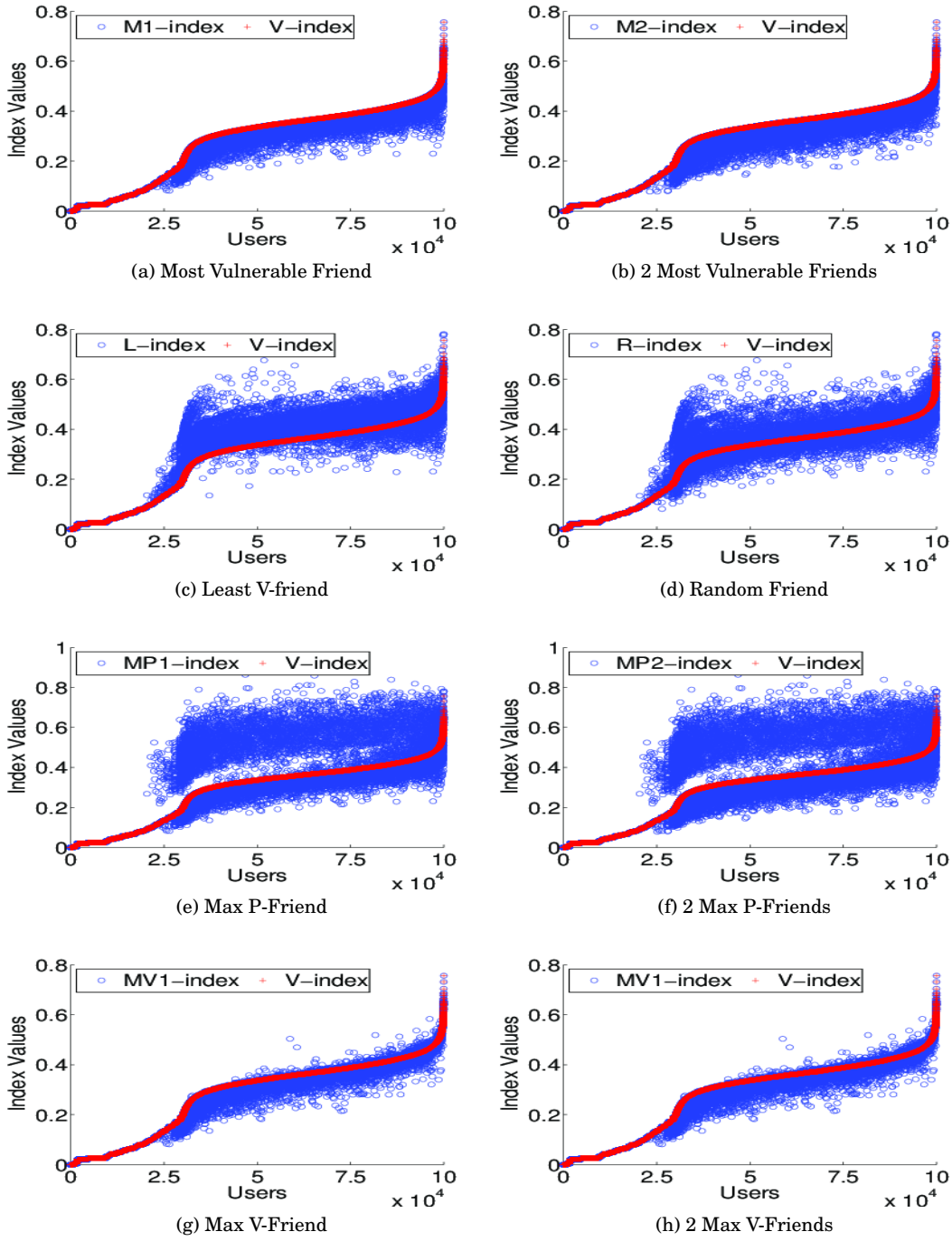
number of Facebook friends, while for Figure 3c, we sort all users in the ascending order based on the number of crawled Facebook friends. Figure 3a shows the relationship between actual Facebook friends (red) and crawled friends (blue) in our dataset. In our experiments, we estimate the V-index of each user as an average of all the P-index values of crawled friends. Based on the V-index, we compute the number of vulnerable friends for each user, as described in Section 3. Figure 3b shows the relationship between actual Facebook friends (red) and their vulnerable friends (blue). Figure 3c shows the relationship between crawled Facebook friends (red) and their vulnerable friends (blue).

5.2. Vulnerability Reduction without Social Utility Constraints

We divide this major task into three experiments to test the (1) impact of different unfriending strategies on users' vulnerability, (2) performance of unfriending the most vulnerable friend with different unfriending strategies, and (3) impact of new friendship on secure users from vulnerable users.

Impact of different unfriending strategies. For the first set of experiments, we compare V-index for each of user with two optimal algorithms and six intuitive strategies for unfriending to reduce vulnerability. For each graph in Figure 4, the X-axis and Y-axis indicate users and their V-index values, respectively. For simplicity, we sort all users in ascending order based on existing V-index, and then we plot their corresponding V-index before and after unfriending. Figure 4 indicates performance of all eight algorithms which will help us to decide whether unfriending makes users more or less vulnerable. The eight algorithms are,

— *Most vulnerable friend*. For a user, the most vulnerable friend is the one whose removal lowers the V-index score the most. For each user, we first find the most vulnerable friend and then estimate the new V-index value (M1-index) after unfriending him/her. As expected, see Figure 4a, V-index values for users decrease in comparison with V-index values before unfriending the most vulnerable friend. Unfriending the most vulnerable friend makes all users more secure.
— *Two most vulnerable friends*. If we sort all of user's vulnerable friends in ascending order based on their new V-indexes (after unfriending), the top two in the list are the two most vulnerable friends. For each user, we first find two most vulnerable friends and then estimate the new V-index value (M2-index) after unfriending them. As expected, see Figure 4b, V-index values for all users decrease in comparison with V-index values before unfriending the two most vulnerable friends. Unfriending the two most vulnerable friends also make all users more secure.



(a) Most Vulnerable Friend

(b) 2 Most Vulnerable Friends

(c) Least V-friend

(d) Random Friend

(e) Max P-Friend

(f) 2 Max P-Friends

(g) Max V-Friend

(h) 2 Max V-Friends

Fig. 4: Performance comparisons of V-indexes for each user before (red) and after (blue) unfriending based on eight different algorithms.

— *Least V-friend*. For each user, we choose to unfriend the friend whose V-index is the lowest among all friends. This friend is the least V-friend. V-index values increase for 65% of 100K users, and increase for 43% of the 2M+ users, in comparison with V-index values before unfriending the least V-friend. See Figure 4c, L-index refers to the new V-index value after unfriending the least V-friend. $V'_u > V_u$ for some users because, $P_l < V_u$ where P_l is the P-index of the least V-friend. Unfriending the least V-friend does not make all users insecure.
— *Random friend*. For each user, we randomly choose to unfriend a friend. V-index values increase for 24% of 100K users, and increase for 23.5% of the 2M+ users, in comparison with V-index values before unfriending a random friend. See Figure 4d, R-index refers to the new V-index value after unfriending a random friend. $V'_u > V_u$ because $P_r < V_u$, where P_r is the P-index of the random friend. Unfriending a random friend does not make all users secure.
— *Max P-friend*. For each user, we choose to unfriend a friend whose P-index is the highest among all friends. V-index values increase for 5% of 100K users, and increase for 11% of the 2M+ users, in comparison with V-index values before unfriending the max P-friend. See Figure 4e, MP1-index refers to the new V-index value after unfriending the max P-friend. $V'_u > V_u$ for some users because $P_{mp_1} < V_u$, where P_{mp_1} is the P-index of the max P-friend. Unfriending the max P-friend makes a majority of users more secure.
— *Two max P-friend*. For each user, we choose to unfriend two friends whose P-index is the highest and second highest among all friends. V-index values increase for 5% of 100K users, and increase for 11% of the 2M+ users, in comparison with V-index values before unfriending the two max P-friends.See Figure 4f, MP2-index refers to the new V-index value after unfriending the two max P-friend. $V'_u > V_u$ for some users because $(P_{mp_1} + P_{mp_2})/2 < V_u$, where P_{mp1_1} and P_{mp_2} are P-indexes of the two max P-friends. Unfriending the two max P-friends makes a majority of users more secure.
— *Max V-friend*. For each user, we choose to unfriend a friend whose V-index is the highest among all friends. V-index values increase for 3.6% of 100K users, and increase for 5% of the 2M+ users, in comparison with V-index values before unfriending max V-friend. See Figure 4g, MV1-index refers to the new V-index value after unfriending the max V-friend. $V'_u > V_u$ for some users because $P_{mv_1} < V_u$, where P_{mv_1} is the P-index of the max V-friend. Unfriending the max V-friend makes a majority of users more secure.
— *Two max V-friend*. For each user, we choose to unfriend two friends whose V-index is the highest and second highest among all friends. V-index values increase for 2.5% of 100K users, and increase for 5% of the 2M+ users, in comparison with V-index values before unfriending the two max V-friends. See Figure 4h, GV2-index refers to the new V-index value after unfriending the two max V-friends. $V'_u > V_u$ for some users because $(P_{mv_1} + P_{mv_2})/2 < V_u$, where P_{mv_1} and P_{mv_2} are P-indexes of the two max V-friends. Unfriending the two max V-friends make a majority of users more secure.

Performance comparison with the best unfriending strategy. In the second set of experiments, we compare the performance of unfriending most vulnerable friends with the seven intuitive unfriending strategies. For each graph in Figure 5, the X-axis and Y-axis indicate users and their associated V-index values after unfriending, respectively. We sort all users in ascending order based on V-index values after unfriending the most vulnerable friend and then plot corresponding V-index based on different unfriending strategies. We find unfriending the most vulnerable friend makes users more secure.

As expected, see Figure 5a-5d, V-index values for each user based on unfriending the least V-friend, a random friend, the max P-friend, or the max V-friend increase for all

users in comparison with their V-index values after unfriending the most vulnerable friend. In the case of unfriending the least V-friend, V-index values increase for 3% of users in comparison with the most vulnerable friend unfriending. Similarly, 1.7% of users increase for a random friend unfriending, 1.7% of users increase for the P-friend unfriend, and 1% of users increase for the V-friend unfriending. Thus, unfriending the most vulnerable friend makes all users more secure than all other schemes.

V-index values for each user based on unfriending the two most vulnerable friends, see Figure 5e, do not decrease for 10% of 100K, and 21% of 2M+ users, in comparison with V-index values after unfriending the most vulnerable friend. V-index values for each user based on unfriending the two max P-friend, see Figure 5f, do not decrease for 51% of 100K, and 81% of 2M+ users, in comparison with V-index values after unfriending the most vulnerable friend. V-index values for each user based on unfriending the two max V-friend, see Figure 5g, do not decrease for 90% of 100K, and 75% of 2M+ users, in comparison with V-index values after unfriending the most vulnerable friend.

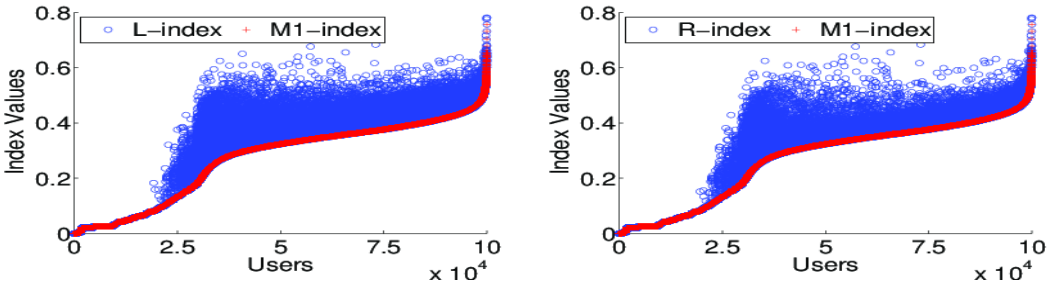
Impact of new friends. We now investigate the impact of new friendship on two types of secure users from vulnerable users. We select three sets of 10K users from 2M+ Facebook users: (S1) users with high V-indexes, (S2) users with low V-indexes, and (S3) C-attributes enabled users with low V-indexes. We randomly select a vulnerable user (i.e., selected from S1, 10K high V-index users) and a secure user (i.e., selected from S2, 10K low V-index users), and pair them and remove the pair from S1 and S2, respectively, until all 10K users from S1 and S2 are paired. We repeat the same with sets S1 and S3. The two sets of results are shown in Figure 6 (a) and (b). For each graph, the X- and Y-axis indicate users and their V-index values before and after the pairing of new friends, respectively. We sort all users in ascending order based on their old V-indexes. As shown in Figure 6a, V-indexes of all users of S2 increase significantly and consistently; in Figure 6b, V-indexes of users of S3 also increase, but vary from minor to large changes. The larger changes in the latter case occur on those users of S3 with fewer friends. The results in Figure 6 confirm that less vulnerable users can become more vulnerable if they are careless when making new friends, and reclusive users are more sensitive to the choice of new friends than less reclusive ones.

5.3. Vulnerability Reduction with Social Utility Constraints

To evaluate our theoretical findings on vulnerability reduction with social utility constraints, we again design three experiments. First, in Figure 7, we compare the V-index value of each user before and after the unfriending of k most vulnerable friends. We refer this as the baseline. We do not consider social utility loss constraint when obtaining the baseline. Second, in Figure 8, we compare the V-index value of each user before and after the unfriending of at most k vulnerable friends while maintaining the acceptable level of social utility loss. We refer this approach as a social utility loss based approach. Third, in Figure 9, we compare the reductions of the baseline and social utility loss approach for each user. We observe from these experiments that it is possible to suggest the maximum number of vulnerable friends to unfriend to achieve desired user vulnerability reduction while maintaining an acceptable level of social utility loss.

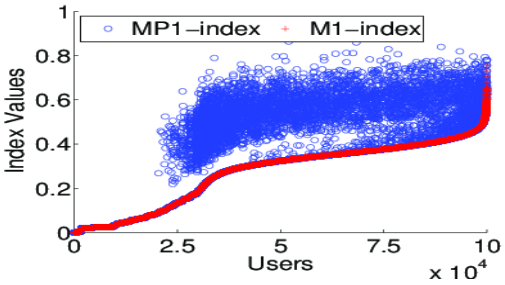
For each graph in Figures 7, 8, and 9, X-axis and Y-axis indicate users and their V-index values, respectively. Without loss of generality, we sort all users, in the ascending order based on the existing V-index values, before we plot the graphs in Figures 7 and 8. For Figure 9, we sort all users in the ascending order based on the V-index values computed using the baseline approach.

The baseline. It consists of results from solving the **VMC** problem with linear objective function $\sigma(\cdot)$. As shown in Section 3, such a problem can be solved in polynomial time. k (or $|D_{u,0}|$, if $|D_{u,0}| < k$) most vulnerable friends are selected for removal to minimize the objective function. We run this experiment on randomly selected 300K users of the

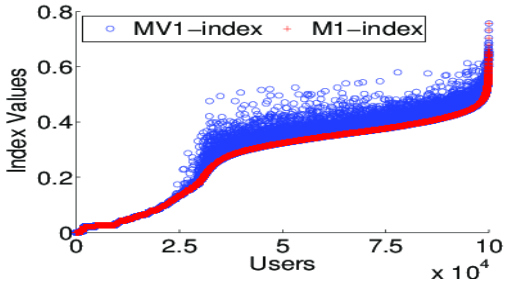


(a) Least V-friend

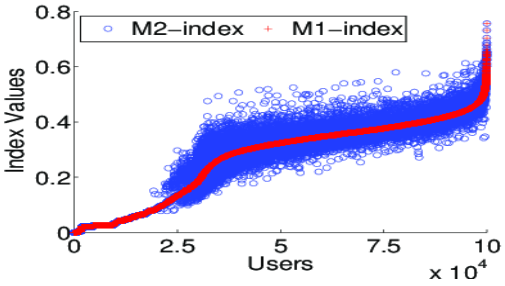
(b) Random Friend



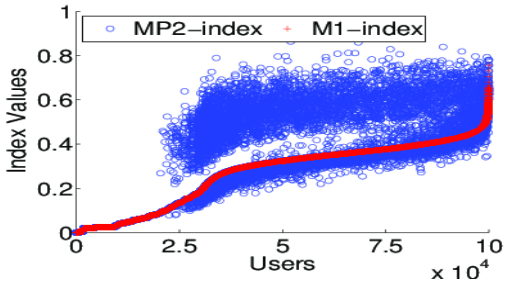
(c) Max P-Friend



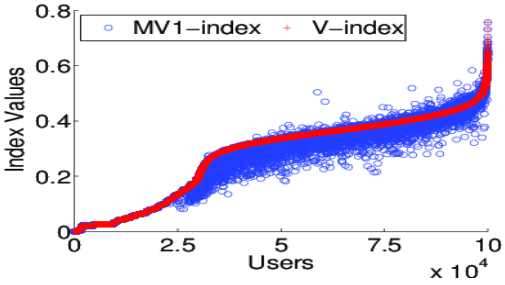
(d) Max V-Friend



(e) 2 Most Vulnerable Friends

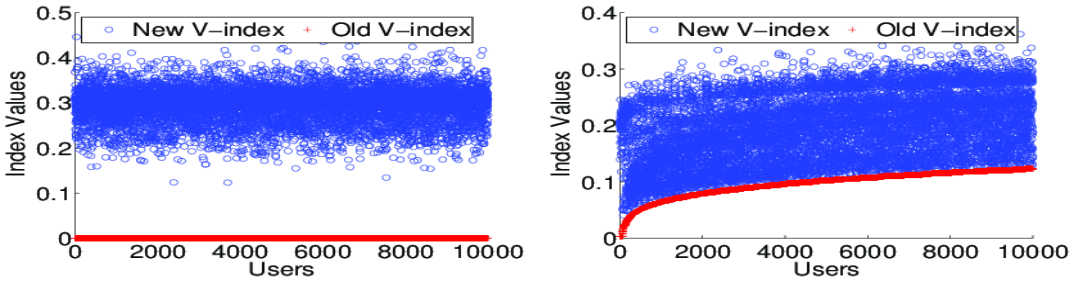


(f) 2 Max P-Friends



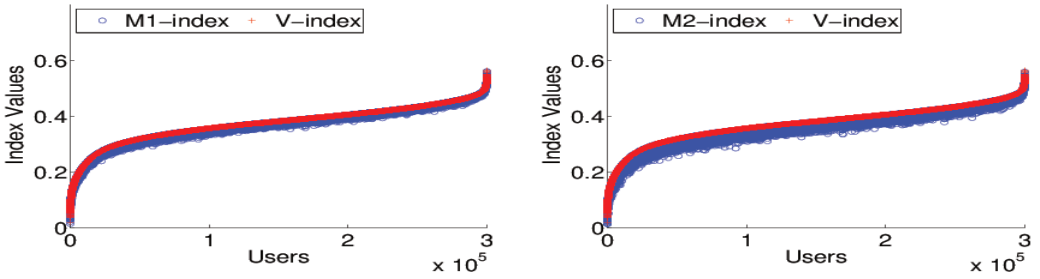
(g) 2 Max V-Friends

Fig. 5: Performance comparisons of unfriending the most vulnerable friend (red) with seven different unfriending ways (blue).

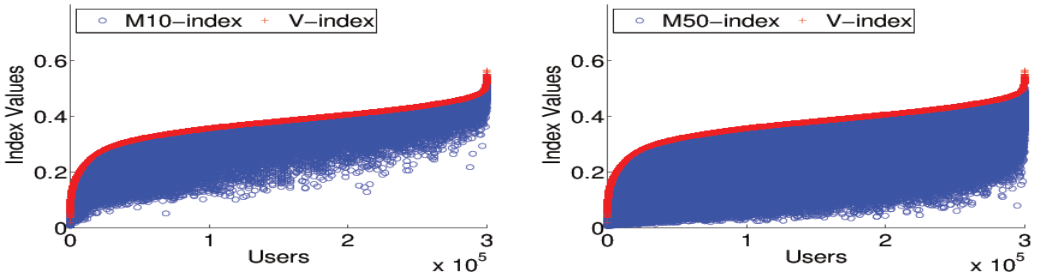


(a) C-attributes are not visible (b) C-attributes are visible

Fig. 6: Impact of new friendship (blue) on users with low V-indexes (red) from users with high V-indexes



(a) Most vulnerable (b) 2 Most vulnerable

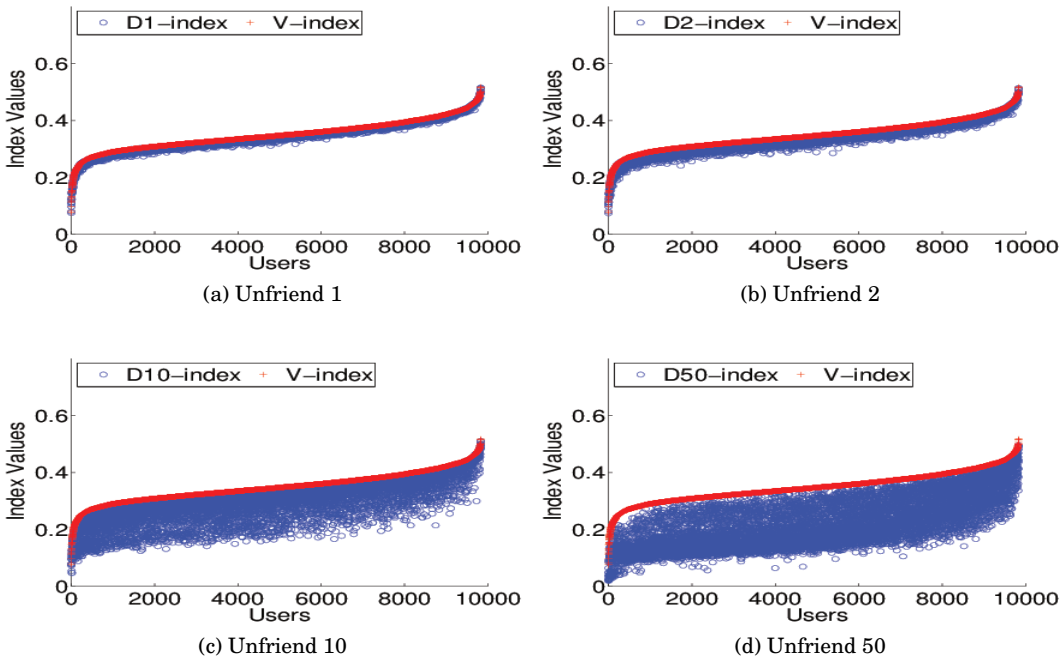


(c) 10 Most vulnerable (d) 50 Most vulnerable

Fig. 7: (The baseline) Performance comparison of V-index values for each user before (red) and after (blue) unfriending the k most vulnerable friends from his social network.

Facebook dataset. Figure 7 shows the performance comparison of V-index values for each user before (red) and after (blue) unfriending at most k vulnerable friends. We run the experiments for different values of k including 1, 2, 10, and 50. As expected, vulnerability decreases consistently as the value of k increases as seen in Figures 7a- 7d. For a given k, the baseline is expected to achieve maximum user vulnerability reduction, but cannot guarantee the retention of socially valuable vulnerable friends.

A social utility based approach. We compute the minimized user vulnerability by solving the **VMUC** problem. For experiments, the V-index for each user is estimated as



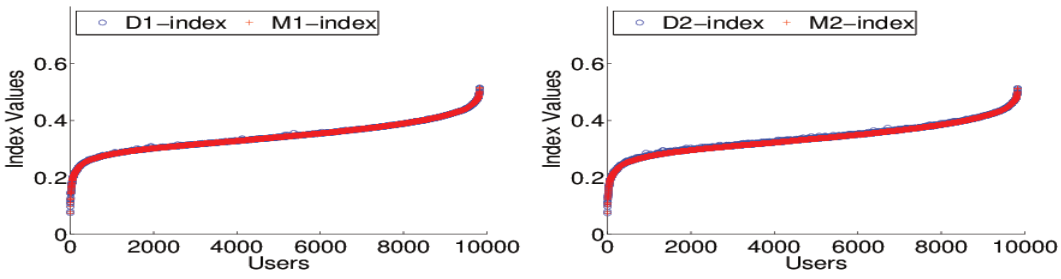
(a) Unfriend 1 (b) Unfriend 2

(c) Unfriend 10 (d) Unfriend 50

Fig. 8: Performance comparisons of V-index values for each user before (red) and after (blue) unfriending at most k vulnerable friends with the total degree of vulnerable users as a social utility constraint. We set error parameter $\epsilon = 0.1$ (input parameter to FPTAS) and retain at least 90% of the total degrees of all the vulnerable friends after unfriending.

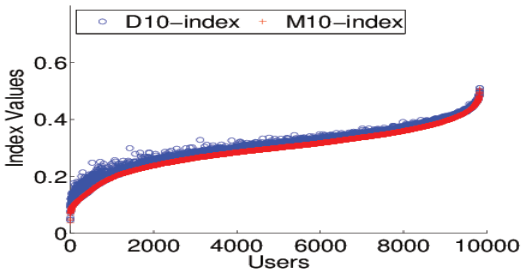
an average of all the P-index values of crawled friends. Hence, the objective function $\sigma(\cdot)$ for **VMUC** is linear. As discussed in Section 4, the scaling and rounding algorithm presented is FPTAS for such a relaxed **VMUC** problem. Figure 8 shows the performance comparisons of minimized V-index values for each user before (red) and after (blue) unfriending at most k vulnerable friends with the sum of all their degrees as a social utility constraint. Due to the theoretical guarantees of FPTAS, we set error parameter ϵ to relatively low value and aim to retain as high number of valuable friends as possible. We set error parameter $\epsilon = 0.1$ (input parameter to FPTAS) and retain at least 90% of the total degrees of all the vulnerable friends after unfriending. As FPTAS runs slower for a smaller error parameter, we run the experiments for randomly selected 10K users out of 300K users. As expected, vulnerability drops consistently as the value of k increases. We show the experiment results for four different k values including 1, 2, 10, and 50 in Figures 8a- 8d. We also run the experiments with other forms of social utility measures such as tie strength and number of common friends. Tie strength between two friends follows a random distribution by having a value between 0 to 1 for each user, where 1 represents the maximum social tie strength. We observe the similar patterns in user vulnerability reduction for these two social utility measures.

Comparing the social utility approach with the baseline. The purpose of this experiment is to evaluate how effective the social utility approach is in reducing vulnerability and retaining social utility. Figure 9 shows the performance comparison. The results for four different k values (1, 2, 10, and 50) are depicted in Figures 9a- 9d. We use the total

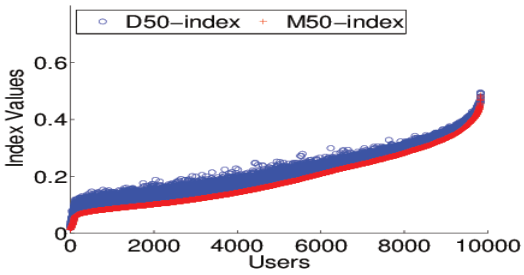


(a) Unfriend 1

(b) Unfriend 2



(c) Unfriend 10



(d) Unfriend 50

Fig. 9: Performance comparison between the baseline and the social utility approach.

Approach	Unfriending k Friends				
	$k = 1$	$k = 2$	$k = 5$	$k = 10$	$k = 50$
Baseline	0.3425 (↓ 1.71%)	0.3371 (↓ 3.26%)	0.3216 (↓ 7.74%)	0.2944 (↓ 15.52%)	0.1891 (↓ 45.73%)
Degree	0.3427 (↓ 1.67%)	0.3381 (↓ 2.98%)	0.3261 (↓ 6.45%)	0.3048 (↓ 12.54%)	0.2185 (↓ 37.31%)
Priority	0.3426 (↓ 1.68%)	0.3378 (↓ 3.07%)	0.3250 (↓ 6.77%)	0.3025 (↓ 13.20%)	0.2146 (↓ 38.43%)
Common Friends	0.3426 (↓ 1.68%)	0.3378 (↓ 3.08%)	0.3249 (↓ 6.8%)	0.3027 (↓ 13.13%)	0.2151 (↓ 38.28%)

Table VI: The average V-index values for the baseline and three different social utility measures. Numbers in bracket are percentage decrease in each average V-index value. The average V-index value of all users before unfriending any vulnerable friend is 0.3485.

number of degrees of vulnerable friends as a social utility measure. As expected, vulnerability reduction for the baseline is more than the social utility approach. However, we can still achieve significant reduction with the social utility constraint. We observe similar results for the other two social utility measures, i.e., tie strength and number of common friends.

Before unfriending any vulnerable friend, the average V-index value of all users is 0.3485. Table VI shows the average V-index values for the baseline and social utility approach. The results for the three different social utility measures are presented. It

also reports the percentage decrease in average V-index value for each approach. The marginal reduction in the average V-index decreases as the value of k increases. Results show that baseline is always the most effective in removing the vulnerable friends. This is because it removes the most vulnerable friends without considering the social utility loss for a given k. But this may incur the cost of loss in social utility value for a user. The social utility approach aims to retain the 90% of user u's social utility value for a given k in when minimizing u's vulnerability. Table VI provides a summary of comparative results for $k = 1, 2, 5, 10, 50$. We observe the following: (1) the more vulnerable friends to remove, the less vulnerable a user is for all four cases (the baseline plus 3 social utility measures); (2) by allowing for a 10% loss of a social utility measure, one can still achieve comparable reduction to the baseline; (3) vulnerability reduction by all three social utility measures are similar; (4) for a given k, the baseline achieves the largest reduction; and (5) but the gain over a social utility measure can be easily eliminated by removing the next larger k friends. For example, unfriending 2 vulnerable friends with any social utility measure can attain vulnerability reduction that is larger than that of the baseline with $k = 1$.

6. CONCLUSIONS AND FUTURE WORK

There are vulnerable friends on social networking sites and it is important to find and unfriend vulnerable friends so that users can improve their privacy and security. However, unfriending vulnerable friends from a user's social network can significantly decrease the user's social utility. In this paper, we study the novel problem of vulnerability reduction with and without social utility loss constraints. First, we provide general model for vulnerability reduction. Using this model, we formulate the two discrete optimization problems, viz., **VMC** and **VMUC**. The **VMC** problem only considers the cardinality constraint while the **VMUC** problem considers cardinality as well as social utility constraints. Both problems are NP-hard. Our experiments on the Facebook dataset evaluate the effectiveness of different methods of vulnerability reduction with and without social utility constraints.

We propose a feasible approach to a novel problem of identifying a user's vulnerable friends on a social networking site. Our work differs from existing work addressing social networking privacy by introducing a vulnerability-centered approach to a user security on a social networking site. On most social networking sites, privacy related efforts have been concentrated on protecting individual attributes only. However, users are often vulnerable through community attributes. Unfriending vulnerable friends can help protect users against the security risks. Based on our study of over 2 million users, we find that users are either not careful or not aware of security and privacy concerns of their friends. Our model clearly highlights the impact of each new friend on a user's privacy. Our approach does not require the structural change of a social networking site and aims to maximally reduce a user's vulnerability while minimizing his social utility loss. The work formulates a novel problem of constrained vulnerability reduction suggests a feasible approach, and demonstrates that the problem of constrained vulnerability reduction is solvable.

An immediate extension is to see whether vulnerability reduction can be performed with multiple social utility measures and if it is cost-effective. More social utility measures are listed in Table III. It would be interesting to study the role of each social utility measure in user vulnerability reduction. Theoretically, the **VMUC** problem for non-linear social utility loss constraint is still open for the future research. We are also interested in investigating the role of user vulnerability across social networks [Zafarani and Liu 2009; 2013] and relationship between the influential user [Agarwal et al. 2008] and vulnerable user while incorporating social theories [Tang et al. 2014].

7. ACKNOWLEDGMENTS

We thank Dr.Gabriel Fung for providing the crawled Facebook dataset for experiments. We also thank the DMML members for their helpful comments. This research was, in part, supported by grants of ARO (025071), ONR (N000141010091, N000141010095) and AFOSR (FA95500810132). This work was also funded, in part, by OSD-T&E (Office of Secretary Defense-Test and Evaluation), DefenseWide/PE0601120D8Z National Defense Education Program (NDEP)/BA-1, Basic Research; SMART Program Office, www.asee.org/fellowships/smart, Grant Number N00244-09-1-0081.

REFERENCES

AGARWAL, N., LIU, H., TANG, L., AND YU, P. 2008. Identifying the influential bloggers in a community. In *the first ACM International Conference on Web Search and Data Mining (WSDM)*.

BACKSTROM, L., HUTTENLOCHER, D., KLEINBERG, J., AND LAN, X. 2006. Group Formation in Large Social Networks: Membership, Growth, and Evolution. In *the 12th ACM SIGKDD*. 44–54.

BADEN, R., BENDER, A., SPRING, N., BHATTACHARJEE, B., AND STARIN, D. 2009. Persona: An online social network with user-defined privacy. *ACM SIGCOMM Computer Communication Review 39,* 4, 135–146.

BECKER, G. 1974. A Theory of Social Interactions.

BOYD, D. M. AND ELLISON, N. B. 2008. Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication 13,* 1, 210–230.

BROCK, W. AND DURLAUF, S. 2001. Discrete Choice with Social Interactions. *The Review of Economic Studies 68,* 2, 235.

EASLEY, D. AND KLEINBERG, J. 2010. *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge Univ Pr.

FANG, L. AND LEFEVRE, K. 2010. Privacy wizards for social networking sites. In *the 19th International World Wide Web Conference (WWW)*.

GAREY, M. AND JOHNSON, D. 1979. *Computers and Intractability: A Guide to the Theory of NP-completeness*. WH Freeman & Co.

GILBERT, E. AND KARAHALIOS, K. 2009. Predicting tie strength with social media. In *the 27th ACM CHI*. 211–220.

GOEL, S., HOFMAN, J. M., AND SIRER, M. I. 2012. Who Does What on the Web: A Large-Scale Study of Browsing Behavior. In *Proceedings of the Sixth International Conference on Weblogs and Social Media*. ICWSM.

GOLBECK, J., ROBLES, C., AND TURNER, K. 2011. Predicting Personality with Social Media. In *CHI Extended Abstracts on Human Factors in Computing Systems*. CHI EA. 253–262.

GOOLSBY, R. 2010. Social media as crisis platform: The future of community maps/crisis maps. *ACM Trans. Intell. Syst. Technol. 1,* 1, 1–11.

GROSS, R. AND ACQUISTI, A. 2005. Information revelation and privacy in online social networks. In *the ACM workshop on Privacy in the electronic society*. ACM, 71–80.

GUNDECHA, P., BARBIER, G., AND LIU, H. 2011. Exploiting Vulnerability to Secure User Privacy on a Social Networking Site. In *the 17th ACM SIGKDD*.

GUNDECHA, P. AND LIU, H. 2012. Mining social media: a brief introduction. *Tutorials in Operations Research 1,* 4.

GUNDECHA, P., RANGANATH, S., FENG, Z., AND LIU, H. 2013. A tool for collecting provenance data in social media. In *Proceedings of the 19th ACM SIGKDD*. ACM, 1462–1465.

HU, J., ZENG, H.-J., LI, H., NIU, C., AND CHEN, Z. 2007. Demographic Prediction based on User's Browsing Behavior. In *Proceedings of the 16th international conference on World Wide Web*. WWW. 151–160.

KOSINSKI, M., STILLWELL, D., AND GRAEPEL, T. 2013. Private Traits and Attributes are Predictable from Digital Records of Human Behavior. *Proceedings of the National Academy of Sciences*.

KRISHNAMURTHY, B. AND WILLS, C. 2010. On the leakage of personally identifiable information via online social networks. *ACM SIGCOMM Computer Communication Review 40,* 1, 112–117.

LESKOVEC, J., ADAMIC, L., AND HUBERMAN, B. 2007. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB) 1,* 1.

LIU, H. AND MAES, P. 2005. Interestmap: Harvesting social network profiles for recommendations. *Beyond Personalization*.

LOEWENSTEIN, G., THOMPSON, L., AND BAZERMAN, M. 1989. Social Utility and Decision Making in Interpersonal Contexts. *Journal of Personality and Social Psychology 57,* 3, 426.

MARCUS, B., MACHILEK, F., AND SCHUTZ, A. 2006. Personality in Cyberspace: Personal Web Sites as Media for Personality Expressions and Impressions. *Journal of Personality and Social Psychology 90,* 6, 1014–1031.

MOROZOV, E. 2011. *The Net Delusion: The Dark Side of Internet Freedom*. Public Affairs.

NARAYANAN, A. AND SHMATIKOV, V. 2008. Robust de-anonymization of large sparse datasets. In *the 29th IEEE Symposium on Security and Privacy*.

NARAYANAN, A. AND SHMATIKOV, V. 2009. De-anonymizing social networks. In *the 30th IEEE Symposium on Security and Privacy*.

NEMHAUSER, G., WOLSEY, L., AND FISHER, M. 1978. An Analysis of Approximations for Maximizing Submodular Set Functions. *Mathematical Programming 14,* 1, 265–294.

PARSONS, T. 1968. Social interaction. *International Encyclopedia of the Social Sciences 7*, 429–441.

QUERCIA, D., KOSINSKI, M., STILLWELL, D., AND CROWCROFT, J. 2011. Our Twitter Profiles, Our Selves: Predicting Personality with Twitter. In *Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom)*. 180–185.

RENTFROW, P. J. AND GOSLING, S. D. 2003. The Do Re Mi's of Everyday Life: The Structure and Personality Correlates of Music Preferences. *Journal of personality and social psychology 84,* 6, 1236–1256.

ROSENBLUM, D. 2007. What anyone can know: The privacy risks of social networking sites. *IEEE Security and Privacy*, 40–49.

SIBONA, C. AND WALCZAK, S. 2011. Unfriending on Facebook: Friend Request and Online/Offline Behavior Analysis. In *the 44th Hawaii International Conference on System Sciences*.

SQUICCIARINI, A., SHEHAB, M., AND PACI, F. 2009. Collective privacy management in social networks. In *the 18th international conference on World wide web (WWW)*.

STERNE, J. 2010. *Social Media Metrics*. The New Rules of Social Media. John Wiley & Sons Inc.

SUBRAMANI, M. R. AND RAJAGOPALAN, B. 2003. Knowledge-sharing and influence in online social networks via viral marketing. *Communications of the ACM 46,* 12, 300–307.

SVIRIDENKO, M. 2004. A Note on Maximizing a Submodular Set Function Subject to a Knapsack Constraint. *Operations Research Letters 32,* 1, 41–43.

TANG, J., CHANG, Y., AND LIU, H. 2014. Mining social media with social theories: A survey. *SIGKDD Explorations*.

TANG, L. AND LIU, H. 2009. Relational learning via latent social dimensions. In *the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*.

TANG, L., LIU, H., ZHANG, J., AND NAZERI, Z. 2008. Community evolution in dynamic multi-mode networks. In *the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*.

TONG, H., PRAKASH, B. A., ELIASSI-RAD, T., FALOUTSOS, M., AND FALOUTSOS, C. 2012. Gelling, and Melting, Large Graphs by Edge Manipulation. In *the 21st ACM international conference on Information and knowledge management*.

TONG, H., PRAKASH, B. A., TSOURAKAKIS, C., ELIASSI-RAD, T., FALOUTSOS, C., AND CHAU, D. H. 2010. On the vulnerability of large graphs. In *the 10th IEEE International Conference on Data Mining*. 1091–1096.

VAYNERCHUK, G. 2009. *Crush It!: Why Now Is the Time to Cash in on Your Passion* 1st Ed. HarperCollins.

VAZIRANI, V. 2001. *Approximation Algorithms*. Springer.

VEBLEN, T. AND BANTA, M. 2007. *The Theory of the Leisure Class*. Oxford University Press, USA.

WATTS, D. AND STROGATZ, S. 1998. Collective Dynamics of Small-world Networks. *Nature 393,* 6684, 440–442.

WONDRACEK, G., HOLZ, T., KIRDA, E., AND KRUEGEL, C. 2010. A practical attack to de-anonymize social network users. In *the 31st IEEE Symposium on Security and Privacy*.

YING, X. AND WU, X. 2008. Randomizing Social Networks: a Spectrum Preserving Approach. In *the SIAM International Conference on Data Mining*.

ZAFARANI, R., ABBASI, M. A., AND LIU, H. 2014. *Social Media Mining: An Introduction*. Cambridge University Press.

ZAFARANI, R. AND LIU, H. 2009. Connecting corresponding identities across communities. In *the 3rd ICWSM*.

ZAFARANI, R. AND LIU, H. 2013. Connecting users across social media sites: a behavioral-modeling approach. In *Proceedings of the 19th ACM SIGKDD*. ACM, 41–49.

ZHELEVA, E. AND GETOOR, L. 2009. To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles. In *the 18th International World Wide Web Conference (WWW)*.