

Fall 2016

Rethinking Privacy and Security Mechanisms in Online Social Networks

Xin Ruan

College of William and Mary - Arts & Sciences, xinruan09@gmail.com

Follow this and additional works at: <https://scholarworks.wm.edu/etd>



Part of the [Computer Sciences Commons](#)

Recommended Citation

Ruan, Xin, "Rethinking Privacy and Security Mechanisms in Online Social Networks" (2016). *Dissertations, Theses, and Masters Projects*. Paper 1499449857.

<http://doi.org/10.21220/S2R07J>

This Dissertation is brought to you for free and open access by the Theses, Dissertations, & Master Projects at W&M ScholarWorks. It has been accepted for inclusion in Dissertations, Theses, and Masters Projects by an authorized administrator of W&M ScholarWorks. For more information, please contact scholarworks@wm.edu.

Rethinking Privacy and Security Mechanisms in Online Social Networks

Xin Ruan

Xi'an, Shaanxi, China

Master of Science, Xidian University 2009
Bachelor of Science, Xidian University 2007

A Dissertation presented to the Graduate Faculty
of the College of William and Mary in Candidacy for the Degree of
Doctor of Philosophy

Department of Computer Science

The College of William and Mary
January 2017

APPROVAL PAGE

This Dissertation is submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy



Xin Ruan

Approved by the Committee, October, 2016



Committee Chair

Professor Haining Wang, Electrical and Computer Engineering
University of Delaware



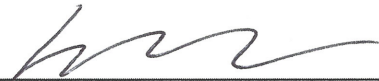
Associate Professor Gang Zhou, Computer Science
The College of William and Mary



Professor Weizhen Mao, Computer Science
The College of William and Mary



Professor Qun Li, Computer Science
The College of William and Mary



Associate Professor Kun Sun, Information Sciences and Technology
George Mason University



Assistant Professor Xia Ning, Computer and Information Science
Indiana University-Purdue University Indianapolis

COMPLIANCE PAGE

Research approved by

Protection of Human Subjects Committee

Protocol number(s): PHSC-2013-08-05-8864

Date(s) of approval: August 9th, 2013

ABSTRACT

With billions of users, Online Social Networks(OSNs) are amongst the largest scale communication applications on the Internet. OSNs enable users to easily access news from local and worldwide, as well as share information publicly and interact with friends. On the negative side, OSNs are also abused by spammers to distribute ads or malicious information, such as scams, fraud, and even manipulate public political opinions. Having achieved significant commercial success with large amount of user information, OSNs do treat the security and privacy of their users seriously and provide several mechanisms to reinforce their account security and information privacy. However, the efficacy of those measures is either not thoroughly validated or in need to be improved. In sight of cyber criminals and potential privacy threats on OSNs, we focus on the evaluations and improvements of OSN user privacy configurations, account security protection mechanisms, and trending topic security in this dissertation.

We first examine the effectiveness of OSN privacy settings on protecting user privacy. Given each privacy configuration, we propose a corresponding scheme to reveal the target user's basic profile and connection information starting from some leaked connections on the user's homepage. Based on the dataset we collected on Facebook, we calculate the privacy exposure in each privacy setting type and measure the accuracy of our privacy inference schemes with different amount of public information. The evaluation results show that (1) a user's private basic profile can be inferred with high accuracy and (2) connections can be revealed in a significant portion based on even a small number of directly leaked connections.

Secondly, we propose a behavioral-profile-based method to detect OSN user account compromisation in a timely manner. Specifically, we propose eight behavioral features to portray a user's social behavior. A user's statistical distributions of those feature values comprise its behavioral profile. Based on the sample data we collected from Facebook, we observe that each user's activities are highly likely to conform to its behavioral profile while two different user's profile tend to diverge from each other, which can be employed for compromisation detection. The evaluation result shows that the more complete and accurate a user's behavioral profile can be built the more accurately compromisation can be detected.

Finally, we investigate the manipulation of OSN trending topics. Based on the dataset we collected from Twitter, we manifest the manipulation of trending and a suspect spamming infrastructure. We then measure how accurately the five factors (popularity, coverage, transmission, potential coverage, and reputation) can predict trending using an SVM classifier. We further study the interaction patterns between authenticated accounts and malicious accounts in trending. At last we demonstrate the threats of compromised accounts and sybil accounts to trending through simulation and discuss countermeasures against trending manipulation.

TABLE OF CONTENTS

Acknowledgements	ii
Dedications	iii
List of Tables	iv
List of Figures	v
Chapter 1. Introduction	1
Chapter 2. Unveiling Intrinsic Breaches of Privacy Settings	10
Chapter 3. Profiling Social Behaviors for Compromised Account Detection	39
Chapter 4. The Security of Twitter Trending	75
Chapter 5. Conclusion	111
References	113
Vita	125

ACKNOWLEDGEMENTS

I would like to express the deepest gratitude to my committee chair, my advisor, Dr. Haining Wang. This dissertation would not have been possible without his guidance. He has been continually encouraging me to explore my research interests and supporting me all through my graduate study.

I would like to express my sincere gratitude to my committee members. Dr. Gang Zhou, Dr. Weizhen Mao and Dr. Qun Li, for their valuable advice on my research and dissertation. Dr. Xia Ning and Dr. Kun Sun, for their support and encouragement as my committee members.

I am thankful to my colleague Yubao Zhang with whom I enjoyed working and completing research projects together. And Dr. Chuan Yue for sharing with me his research experience and giving me helpful research advice.

I am grateful to the staff of the Department of Computer Science. Especially Ms. Vanessa Godwin and Ms. Dale Hayes for their kind help and unfailing administrative support.

To my husband, Adam, and to my daughter, Amanda and to my parents, for their
love, encouragement and support.

LIST OF TABLES

1. User Sets and Ratio	19
2. Feature Value Comparison	54
3. A Behavioral Profile Sample	57
4. The Jensen-Shannon Divergence	82
5. Topics Extracted from Datasets	93

LIST OF FIGURES

1. User Information from An Attacker's View	11
2. Number of Leaked Friends	22
3. Number of Correctly Inferred Attribute Values	29
4. Inference Accuracy	30
5. Top Institutions Accuracy	31
6. Community Feature Sharing	32
7. Friends in the largest Component	34
8. Traversed Friends Ratio in 1 Hop	35
9. Traversed Friends Ratio in 2 Hops	35
10. Private-Friends Inference Ratio	36
11. Facebook Data Set Overview Statistics	49
12. Combined Distributions of Extroversive Features	51
13. Combined Distributions of Introversive Features	53
14. Behavior Profile Difference & Variance	63
15. Impact of Training Data Size	66
16. Impact of Feature Quality	68
17. Impact of Profile Completeness	69
18. Coverage and Mean Position of Sample Trends	80
19. Sample and Search Dynamics	81
20. Observed and Estimated Dynamics of "ThrowbackThursday"	85

21. Normalized Number of Follower and Retweet for “ThrowbackThursday”	86
22. Waiting Time of Accounts in the Spike	88
23. Average Descendant Number of Malicious Accounts	89
24. Example of Kalman Filter	91
25. Example of One Segment	92
26. Trending Duration of Topics	94
27. The Best Accuracy for Dynamics of Each Factor with $M = 12$	97
28. Variation of Segment Size ($M \in \{4,8,12,16\}$)	98
29. Correlation of Suspended and Authenticated Account Dynamics	99
30. Best Accuracy of Suspended and Authenticated Account Dynamics	100
31. Malicious and Authenticated Account Peaks for the Topics	101
32. Avg. Follower and Tweet History Number of Spammer and Descendants	102
33. Entropy and Retweet Rate of the Accounts	103
34. Ratio of Friend Number to Follower Number	105
35. Ratio of Predicted Trends	106

Chapter 1

Introduction

With over one billion users, Online Social Networks (OSNs) are amongst the largest scale communication applications on the Internet. As of the second quarter of 2016, the number of monthly active users on Facebook reached more than 1.6 billion [8]. Launched later than Facebook, Twitter has more than 200 million monthly active users [20]. OSNs have changed multiple aspects of our daily lives by providing a great variety of social communication services. OSNs enable users to easily access local and world-wide news, as well as share information publicly and interact with friends. Furthermore, celebrities leverage OSNs to attract attentions and gain more popularity; company and organizations set up pages on OSNs for advertising and marketing. On the negative side, OSNs have also been abused by spammers to distribute ads or malicious information [16], such as scams, fraud, and even manipulating public political opinions [93].

One of the top concerns of OSN users is their privacy and security [17]. In exchange for the convenience of the social communication services, users entrust OSNs with their personal information. The massive amount of users and their personal information, in turn, help OSNs to generate profit through marketing services, such as targeted advertisements [11]. Having achieved significant commercial success [18]

with this mode, OSNs do take the security of user private information into serious consideration. The major OSNs provide their users with extensive privacy configurations, which allow fine-grained control of private information visibility to specific friends and strangers. In addition, most OSNs employ two factor authentication [13], IP geo-location monitoring [3], and account recovery processes [15] to boost user account security. Moreover, to help user to consume information more conveniently, OSNs publish the most hot topics in real time. However, the efficacy and security of these measures are either not thoroughly studied or in need to be improved.

In sight of cyber criminals and potential privacy threats on OSNs, serious research efforts have been paid on related topics. A large portion of OSN security research in recent years has focused on spammer accounts and spam analysis . Spam detection methods used for emails are introduced to OSNs [54]; and then new methods leveraging OSN features, such as user connections, are also proposed later [47, 52, 108]. Because spams are usually post automatically from bot accounts, botnet detection also helps to detect spammers [102, 111]. Sybil detection is also well studied to discern large groups of fake OSN accounts [37, 43, 98]. Different approaches [82, 103, 110] have been proposed to evaluate account reputation or vouch for accounts; and defend large scale crawling on OSNs [56, 74]. Moreover, [40, 64, 72, 81, 112] have revealed user security and privacy breaches on OSN by collecting and inferring user information while [31, 46] have redesigned OSN system structure to reinforce users' privacy.

This dissertation focuses on the evaluations and improvements of OSN user privacy configurations, account security protection mechanisms and trending topic security. More specifically, for the user privacy configuration, we identify an information leakage vulnerability in major OSNs, which is mainly caused by incomprehensive privacy policy coverage. We conduct an extensive measurement study on Facebook, and present the quantity and quality of private information an attacker could re-

veal due to this vulnerability. For account security protection, we target at account compromise, which is an emerging prevalent and significant problem due to the difficulties in accurate detection and effective mitigation. We devise a novel account activity anomaly detection mechanism based on passive observation of users' social behaviors. By conducting measurement studies on more than 50 human users of their Facebook usages, we show that users' social behaviors on OSNs are diverse and distinct, and thus can be leveraged to efficiently identify account compromise. For trending topic security, we investigate the manipulation of trends in Twitter, a popular OSN. As Twitter become one of the major ways for information consuming, many OSNs list the most hot topics to users in real time. Given the millions of tweets we collected in Twitter, we evidence the manipulation of trending topics by malicious accounts. To explore how they achieve that, we study which factor impact trending the most. We further demonstrate the threat of malicious manipulation of Twitter trending by using simulation based on the spamming infrastructure we observed.

1.1 Breaches in Privacy Settings

Major OSNs, including Facebook and Google+, strive to protect their users' privacy by extensive privacy configurations. However, how secure a user's private information is with these protection mechanisms has not been thoroughly investigated. With the privacy settings, a user is given the impression that it has total control over the visibility of each piece of its private information. Unfortunately, as revealed in Chapter 2, many of the privacy settings on OSNs are ineffective, and the protection users enabled on their private information can be easily bypassed.

Although the granularity varies on different OSNs, a user's personal information is usually categorized into several different classes, such as identity (e.g., name, birthday,

and photos), community (e.g., location, and organization affiliation), relationship (e.g., friend list and follower/following connections), etc. Users could independently configure the publicity levels (e.g., visibility to general public, to groups of users, and even to specific users) of each class of information. However, the seemingly extensive set of privacy settings often gives users false sense of security, because the configurations are enforced by incomprehensive, and sometimes even conflicting privacy policies. For example, if a user chooses not to show its friend list to strangers but allows everyone to view the photo albums, its friends are being “leaked” by the comments on the photos.

Existing research on privacy settings focuses on the diverge between users’ privacy expectation and their privacy settings [68, 70]. However, the previous studies assume that privacy settings would provide proper protection to the corresponding information, and have not considered the intrinsic vulnerabilities of incomprehensive privacy policies. While many previous research efforts on privacy setting breaches strive to infer user information [40, 64, 72, 112], none of them take into account the effects of privacy settings over information availability. And additionally, these studies suffer from the limitations of overly powerful attackers (e.g., assuming the availability of thousands of users as the training dataset [112]), small scopes (e.g., only on specific kinds of information such as music interests [40] and group membership [64]), and coarse granularity (e.g., infer information of user groups, instead of individual users).

We investigate the ineffectiveness of the OSN privacy configurations from an attacker’s point of view. Using a large-scale measurement study on Facebook, we quantify the prevalence of unintentional private information leakages among users. Then, considering ourselves as attackers who have no connection to victim users, we design sophisticated inference algorithms that adapt to the victims’ privacy configurations and public information availability, and exploit the privacy policy loop-holes to infer

unpublished personal information of the victims. Our evaluation shows that, without the needs of a large training dataset, our attack schemes can uncover a remarkable amount of user private information with high accuracy. This indicates that the information leakage vulnerability on today’s OSN can lead to significant privacy breaches.

1.2 Compromised Account Detection

With the increasing popularity of OSNs, account compromise has emerged as a significant threat to OSN users. Recent news have reported that Twitter accounts were hacked on a large scale [1], and in addition, both Thomson Reuters’ Twitter account and Facebook CEO’s account were hijacked [22, 24]. These incidents evident that today’s OSNs lack adequate protections to their users’ accounts.

Compared to sybil accounts and dedicated dummy accounts for conducting malicious activities, compromised accounts are more favored by cyber criminals. On one hand, the pre-existing social connections between the victim users and their friends could be exploited to distribute malicious information (such as spams) more effectively. On the other hand, the well established trust relationship between the service providers and account owners makes the detection of compromised accounts quite challenging. Previous research on offline spam analysis shows that most spam emails are distributed via compromised accounts [48, 52].

One method OSN employ to prevent account abuse is the two factor authentication, which requires more steps than normal login process and affects user experience. Other mechanisms adopted to discern abnormal account behavior are based on geolocation and browser information [3, 5], which could be evaded by hackers. Thus, more effective solutions are needed to detect compromised accounts. Based on individual user’s online social behaviors, we propose to detect compromised accounts

by building a behavioral profile for the authentic user of an account. The rationale behind our design is that the disparity to the user profile indicates compromise.

Compromised account detection has to be differentiated from spammer account detection, though most existing research has not yet done so. While dedicated spammers can be banned directly upon detection, compromised accounts cannot be simply banned due to the fact that those accounts are actually common users' and banning those accounts affects user experience. Most existing research on spam account detection does not discriminate compromised accounts [47, 48, 52, 105]. One previous work that is specifically dedicated to compromised account detection is based on message posting features and message content clustering [44]. Considering a large number of real-time messages, the method in [44] employing message clustering introduces significant overhead. Therefore, we seek an alternative solution that avoid examining the message content to discern compromised accounts.

Without analyzing user profile or message content, we attempt to discern behavioral anomaly of compromised accounts based on their original users' social activity patterns. The user behaviors can be observed via clickstreams in a lightweight manner. OSNs provide various social services, including browsing friends' updates, uploading photos, and publishing messages, etc. A user's usage pattern in those services depends on its interests and social habits. Hence, the behavioral patterns using those services vary among different users. Meanwhile, the patterns are hard to obtain and arduous to feign.

Based on previous intuition and reasoning, we propose to profile user behavioral pattern as a reference to detect compromised account. After collecting several sample users' clickstreams to Facebook, we study their social behaviors. Based on our observation of their interaction with different OSN services, we propose several new behavioral features to portray user behavior patterns. Our preliminary measure-

ment results on the features validate their effectiveness to quantify user difference. For each behavioral feature, a metric is derived as the statistical distribution of the feature value range, observed from a user’s clickstreams. Combining respective behavioral metrics, a user’s behavioral profile is built and represents the user’s behavioral patterns. In our evaluation, a user’s behavioral profile is employed to differentiate clickstreams of the user from all other users’, and cross-validation experiments are conducted. The evaluation results show that social behavioral profile can effectively differentiate individual OSN users with accuracy up to 97%.

1.3 Trending Topic Manipulation

With the ever-increasing popularity of OSNs, they are not only convenient platforms for communication but also important resource to retrieve news and information [4]. Some events originated from OSNs have become phenomenal [21]. To help users to get news easier, OSNs usually rank the hot topics and update the list in real time. Those trending topics are usually listed in users’ homepage for available retrieval. Given this handy feature, users can easily get access to the most populous trends and join them. Journalists can also take advantage of this feature, and Twitter, Google, Instagram even become important resources for them to develop stories, track breaking news and investigate public opinions. For example, in an election campaign Twitter trends were tracked to learn candidates’ popularity and predict the election outcome [58].

Though trend lists in OSNs facilitate information propagation, they can easily become subject to abuse. Trend ranking algorithms differ among various OSNs, but they are closely related to the trending topic popularity. For instance, Google Hot Trends ranks the latest topics that experienced large surge of search on its site, and the ranking of a topic can be enhanced by manipulating bots to search for the topic on

Google repeatedly. In Twitter, a topic can be promoted in trending by manipulating sybil accounts to mention it in tweets. An article [23] in Wall Street Journal reveals that there are underground markets to promote topics to trending. Moreover, [58] discloses that Twitter have been manipulated in an election campaign.

As there are increasing emphasis on OSN trends, lots of related research have been conducted. To detect trending topic in a timely manner from large amount of information, different trends detection algorithms [25, 38, 59, 69] have been proposed. In addition, Becker *et al.* [34] use a clustering method to distinguish real-world events from trends. To classify various trends, [55, 63, 76] have proposed different methods for trending topic taxonomy. Furthermore, [32, 39, 84, 99] have proposed several means to measure the influence of trending topics and related users. However, there is little research dedicated to trending topic manipulation.

We aim to investigate the abuse of trending topics in Twitter. We attempt to know whether malicious users can manipulate Twitter trends and how can they achieve it. Being exposed to the hottest trends, users are entitled to gain insight into the authentic popularity of the topics. Meanwhile, exploring the manipulation of Twitter trends helps to understand the promotion of a topic from the perspective of a third party. Based on millions of tweets we collected from Twitter, we first evidence the manipulation of Twitter trends by employing an influence model. Then we disclose a suspect spamming infrastructure after the analysis of accounts in the spike of a trend. Given five factors that may affect trending, we study the extent to which a factor can affect trending using an SVM classifier. Moreover, we demonstrate the threat of malicious manipulation of Twitter trending through simulation and discuss the corresponding countermeasures.

1.4 Organization

The remainder of this dissertation is structured as follows. In Chapter 2, we present our proposed algorithms to unveil user privacy and evaluate the efficacy of privacy settings. In Chapter 3, we present our novel social-behavior-based method for compromised account detection. In Chapter 4, we investigate the manipulation of trending topics in Twitter. Finally, we conclude this dissertation in Chapter 5.

Unveiling Intrinsic Breaches of Privacy Settings

Users of online social networks (OSNs) share personal information with their peers. To manage the access to one's personal information, each user is enabled to configure its privacy settings. However, even though users are able to customize the privacy of their homepages, their private information could still be compromised by an attacker by exploiting their own and their friends' public profiles. In this chapter, we investigate the unintentional privacy disclosure of an OSN user even with the protection of privacy setting. We collect more than 300,000 Facebook users' public information and assess their measurable privacy settings. Given only a user's public information, we propose strategies to uncover the user's private basic profile or connection information, respectively, and then quantify the possible privacy leakage by applying the proposed schemes to the real user data. We observe that although the majority of users configure their basic profiles or friend lists as private, their basic profiles can be inferred with high accuracy, and a significant portion of their friends can also be uncovered via their public information.

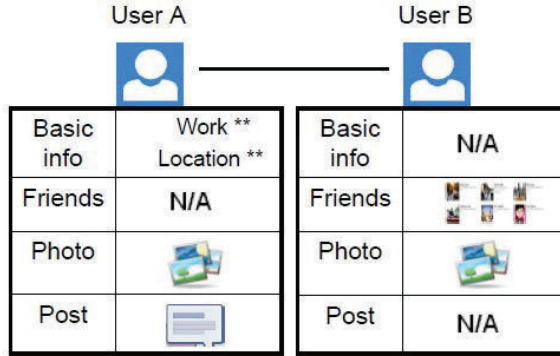


Figure 2.1: User Information from An Attacker’s View

2.1 Motivation

Online social network (OSN) websites have attracted a large number of users in the past few years. Facebook, the most popular OSN, was launched in 2004; by March 2013, the monthly active users exceeded 700 million [7]. Each user account typically includes the user’s basic profile, such as gender, education, and friend list, and other personal data, such as photos and messages. Clearly not every user is willing to share all its information with peer users, either friends or strangers [71]. Accordingly, many social network sites allow a user to take control over its information visibility by configuring privacy settings. Thus, users are able to set their information visibility to different types, and the setting granularity varies from site to site. For instance, except for profile image and name, a Facebook user is capable of configuring its friend list, each piece of profile information, wall post and photo accessibility to strangers and specific friends.

However, some of an OSN user’s private information that is protected by its privacy setting can be easily compromised. In other words, a privacy setting is not effective as what it claims to be. This is due to the intrinsic vulnerabilities inside the privacy setting policy. For instance, as shown in Figure 2.1, user A and user B are mutual friends; each configures its privacy independently such that their information

visibility are as the figure shows. An attacker, who does not set up connections with user A or user B, has no access to user A's friend list, but can access some of its photos or posts; thus some of user A's friends, who responded to A's posts or left photo comments, are leaked. When the attacker also visits user B, who has a public friend list, the attacker can confirm the connection between A and B. Exploiting this kind of vulnerability, we wonder whether A's friends or B's basic information could be uncovered even with the protection of their personalized privacy settings. More generally, we attempt to measure, from an average attacker's perspective, with limited resources, how much of a user's privacy could possibly be compromised based on its plainly leaked information.

From the stance of a stranger to a target user, this chapter strives to evaluate the user's privacy setting breaches on a large scale and attempts to answer the following questions:

- Can one's privacy setting be undermined by developing more sophisticated and practical schemes, which can infer more private profile information based on what has been directly published from the person's homepage?
- How accurate can users' privacy be inferred? While users can configure their privacy settings to different types, can the amount of inferred privacy be quantified given each privacy setting type?
- Is the amount of inferrable privacy mainly determined by the user's privacy setting? If so, can the number of affected users with a certain setting be estimated on a large scale?

Although previous research [68, 70] has investigated the gap between OSN users' privacy expectation and their actual privacy settings, the vulnerabilities in privacy

settings themselves are not studied. Yet there are rare existing research that specifically examines whether a privacy setting can keep the privacy of user information as it is configured. While several efforts [40, 64, 112] have demonstrated the possibility to infer OSN users' one attribute value from another, or to infer the connections, they are based on (1) a large amount of training data [112] or (2) the assumption of the availability of specific kinds of information, such as group membership [64, 112] and music interests [40], which in reality may be set as private by users. The effects of users' privacy settings upon their profiles are not taken into account, let alone to measure the privacy setting breach. A large number of users, who share certain attribute values with the target users, are required as the training data to conduct the information inference. Thus, those strategies can only be taken by attackers with rich resources.

In this chapter, we investigate whether certain privacy settings can effectively protect a user's private information as the user configured. We dwell on measuring and quantifying the unintentional leakage of a target user's basic profile information and friend list, which are the pivot of its social profile. For each target user with a certain privacy configuration, we propose the profile and connection inference schemes based on the user's publicly available information. In addition, instead of relying on a large amount of training data, our approach only needs a small number of users in the target user's neighborhood. The proposed schemes can be conducted by any average users without many resources. We crawl and collect about 300,000 Facebook users' publicly available information as our dataset. The status-quo of those users' privacy settings is measured. Then, we quantify the amount of inferrable private information by using our proposed schemes, and observe that a remarkable amount of privacy could be uncovered, indicating that privacy settings do not effectively guarantee users' information privacy.

2.2 Related Work

There are two major research directions on the privacy and security issues in OSNs: (1) to reveal the privacy threats in OSNs by conducting surveys [68, 70] and proposing attack models [101], information inference algorithms [33, 40, 45, 60, 64, 72, 109], de-anonymization algorithms [30, 77], and re-identification algorithms [108]; and (2) to reinforce users' privacy by redesigning the OSN system structure [31, 46, 73, 87] and conducting anonymization [80, 91]. This chapter investigates the privacy setting breaches, which belongs to (1). We describe the related work as follows.

The disparity between users' actual privacy settings and their privacy expectation in Facebook has been studied by Madejski et al. [70] and Liu et al. [68]. They obtained users' expectations by conducting surveys and retrieved their factual privacy settings; and then detected the inconsistency between the two. Both found that there was a significant variance between users' privacy expectations and their privacy settings. But this is due to users' inability to configure their privacy settings according to their will, and they assumed that the privacy setting could effectively protect the data that it is configured to protect. In contrast, this chapter intends to challenge this assumption and unveils the privacy setting vulnerability in itself. In addition, we measure the privacy setting status-quo on a much larger scale.

Regarding information inference, there are profile mining [33, 40, 72, 112] and link mining [60, 64, 65, 89, 109] approaches, both of which this chapter explores. Zheleva et al. [112] presented several classification models using links and group memberships to infer the target users' profiles. But in many OSNs such as Facebook, the group membership is covert by default. Moreover, it assumes that a specific percentage of attribute values are publicly available to perform the inference, and a user set that consists of thousands of users as training data is needed for classification.

Chaabane et al. [40] extracted semantic correlations among users' music interests, and computed each user's probability vector belonging to certain semantic topics. Users with similar vectors shared the same attribute value. However, this method is limited to users who have published music interests, and is not applicable to more general users who have not done so. A large dataset is also needed for classification.

Mislove et al. [72] assumed that users sharing the same attribute values were inclined to form dense communities. The traditional community detection algorithm is modified to take user's attribute values into consideration. The algorithm is applied to a school student dataset to infer their majors schools, and etc., but when it is applied to a larger user set from a broader geographical area, the accuracy is much lower than that using the student dataset.

Compared to these related work, this chapter designs inference schemes from the stance of an individual user instead of a global view, thus it avoids the need of large amount of training data and only demands the information of the target user's reachable neighbors. More importantly, we take the actual availability of user information into account, instead of assuming specific attribute values to be in hand.

Another important privacy threat is the compromise of a user's connections, i.e., the friend list. Leroy et al. [64] uncovered the social graph given the user's group membership information. However, it is not easy to obtain these group-related data in most OSNs, in which group information is private. Staddon et al. [89] inferred a user's friend list based on the situation that most OSNs provide the shared friend function once a connection has been set up to the target user. However, the dilemma is if the attacker connects to the target user, likely the target user's friend list is already accessible to the attacker. Bonneau et al. [36] also aimed at uncovering a target user's friend list in Facebook by exploiting the public listing feature, but the feature has been disabled and is not available anymore.

2.3 The Facebook Dataset

Facebook was chosen as our research target because it is the world’s most populous OSN providing many flexible features and diverse user resources. More importantly, its privacy setting policy is similar to the policies that most existing OSNs adopt, but in finer granularity. In Facebook, one can set each of its information item individually to be visible to every user (a.k.a. “Public”), or visible only to specific or all friends.

While collecting the dataset, the collector acts as a user who neither belongs to any specific group nor sets up connections with any of the sample users. The retrieved data are all set as “Public,” i.e., accessible to every normal user. Hence, the inference experiments can be reproduced by any other users. Moreover, since we only collected public information, none of Facebook’s security policies were broken. For privacy concern, user names and IDs are anonymized.

The dataset is organized into a database, consisting of about 300,000 Facebook users. The crawling originated from 50 graduate students at the same institution and was conducted in a breadth-first manner. Out of the total users, about 120,000 users were crawled at the beginning phase, and all their main profile subpages were collected. The rest about 180,000 users were crawled thereafter, and all but their photo subpages were collected as photo pages are not used for evaluation. Out of the 300,000 users, there are 909 users all of whose friends’ profiles are also in the dataset; for the rest of users, only some of their friends are in the dataset.

To quantify the information leakage, we emphasize the unintentional revelation of a user’s *targetProfile*, including an attribute set: {location, institution} and the friend list. The attribute set is called the basic attribute set, and its element is basic attribute. While *targetProfile* is the pivot of a user’s social profile, other information items from wall like status, messages, to photos are not included in it because they

are improvised and hard to infer. We define the percentage of users that have certain information public as “public ratio.”

Based on our dataset, the public ratios of users’ four main subpages are: 83.8% for profile page, 62.2% for friends page, 55.1% for wall page, and 45.6% for photo page. For a profile page, it is considered to be public if at least one value in the basic attribute set is visible. A photo or wall page is considered to be public if at least one album or post is visible; A friend list is considered public when it is visible.

As many as 37.8% of users conceal their friend lists from strangers. Compared to about 28% for the dataset in Gundecha’s work [53], more users in our dataset are aware of connection privacy. Although about 83.8% of users publish one or more basic attribute values, a majority of them provide incomplete basic profiles. Based on the dataset, only 9.9% of users publish complete basic attribute values.

Those statistics demonstrate that a significant number of users customize their *targetProfiles* as private or partially private. The inference of their *targetProfiles* reflects the effectiveness of their privacy settings. Next, we present the schemes to infer each of the two *targetProfile* items in detail.

2.4 Exploiting Privacy Setting Vulnerability

Targeting a user’s *targetProfile*, we design different inference schemes for each possible privacy setting type on the four subpages, including profile, friends, wall, and photo. For easy presentation, the notations we used are listed as follows:

U : user set.

$PS(u)$: $u \in U$, user u ’s privacy setting on four subpages: profile, friends, wall, photo in sequence; denoted as a 4-tuple, and entry value 1 means all basic attributes

are visible in the profile page, visible friend page, some visible posts on the wall or photos, respectively, while 0 represents the opposite.

$BA(u) : u \in U$, user u 's basic attribute values.

$FL(u) : u \in U$, all users in u 's friend list, denoted as a user set.

$targetProfile(u) : u \in U$, user u 's *targetProfile*, that is $\{BA(u), FL(u)\}$.

$G = (V, E) : the social graph formed by users in user set V , and E consists of the undirectional connections among users in V ; $\forall u, v \in V$, if $v \in FL(u)$ and $u \in FL(v)$, $(u, v) \in E$. Most frequently it is used to denote a user's neighborhood graph.$

$GC(k) : 1 \leq k \leq n$, a set of members of a community structure detected in a user's neighborhood, and n communities detected in total.

The scenarios under which the *targetProfile* has to be inferred include when $PS = (0, 1, x, x)$, $PS = (1, 0, x, x)$ and $PS = (0, 0, x, x)$, where x can be either 1 or 0. According to the inference objective and public information, we categorize users into four sets from $U1$ to $U4$ by their PS values. $U1$ and $U2$ consist of users whose BA values can be inferred while $U3$ consists of users whose FL can be inferred from their public information, and $U4$ consists of those whose BA or FL are hard to be directly inferred from their public information.

Table 2.1 shows the possible PS values in each user set and the ratio of users in it. About 8.2% of users display complete *targetProfiles* to strangers, thus they are not the inference objects. The union of $U1$, $U2$ and $U3$ consists of 69.4% of users, those users' *targetProfiles* are not complete with more or less additional information accessible. In the following subsections, we first illustrate BA inference followed by

Table 2.1: User Sets and Ratio

User Set	$U1$	$U2$	$U3$		$U4$	
PS	0100	0001	0001	1001	0000	11xx
	0101	0010	0010	1010	1000	
	0110	0011	0011	1011		
	0111					
Ratio	54.0%	14.3%	15.4%		22.4%	8.2%

FL ; in particular, we infer BA for users in $U1$ and $U2$, then we infer FL for users in $U3$, followed by the hardest case for users in $U4$.

2.4.1 Basic Attributes from Friends

The users in $U1$ display incomplete or no BA but their friend lists are visible, and their BAs should be inferred. Table 2.1 shows that 54% of users belong to $U1$, indicating that a large group of users' privacy are threatened if their BAs can be properly compromised. This scenario is formulated as:

$$U1 = \{v | v \in U \text{ and } PS(v) = (0, 1, x, x)\};$$

Inference objective: $BA(v), v \in U1$;

Public information: $FL(v), v \in U1$.

Intuitively, a user's geographical location, occupation, and interests affect the formation of its social circle. Some connections are set up with colleagues or classmates, while others are from interest communities. Thus, its friends could be classified into different groups, each of which is distinguished by an attribute value shared by the group members and the user. Some of its friends may belong to multiple groups. For example, one author's Facebook friends can be classified into three main groups: one from college, one from graduate school, and one from the current city. Some friends from the graduate school are also in the current city, while no one from college is in

the current city. The three groups are distinguished by attribute values at the city or institution level. The friends could be classified into smaller groups by using finer granularity attributes like class and department. The friends in the same group have a higher chance to connect to each other than those from different groups. In other words, community structure exists in the user’s friend circle: the connections inside a community are denser than the connections among communities [50].

Therefore, for $v \in U1$, this feature can be exploited to infer $BA(v)$, i.e., to study the connections among v ’s neighbors and detect communities. We first obtain the social graph in v ’s neighborhood, $G = (V, E)$ and $V = FL(v)$, by traversing v ’s friends and retrieving their profile pages and friend lists, although some of them are private. Then, we conduct the community detection in the graph. After that, we identify the most widely shared basic attribute value within each community as the *community feature*, and assemble those features together to form $BA(v)$. During the neighborhood traversal, neither users who have private profiles nor those who have private friend lists are eliminated during the process. This is because their information could be leaked from their shared friends with v , who have looser privacy configurations. The steps to infer $BA(v)$ are detailed below as **Scheme 1**:

1. Traverse each user $u \in FL(v)$ and retrieve $BA(u)$ and $FL(u)$; form v ’s neighborhood graph $G = (V, E)$, $V = FL(v)$, based on $FL(u)$ for each $u \in FL(v)$.
2. Detect the communities in v ’s neighborhood graph, $G = (V, E)$, $V = FL(v)$, using Girvan-Newman algorithm [50]; and the resulting communities are denoted as $GC(1), GC(2), \dots, GC(n)$.
3. For each community $GC(k)$, $1 \leq k \leq n$, find the *community feature* $A(k)$ and its frequency such that $A(k) \in BA(u)$ for $u \in GC(k)$ and $A(k)$ is the most widely shared basic attribute value among the community members.

4. Merge $A(k)$ and sum up their frequencies for $1 \leq k \leq n$; then sort the merged $A(k)$ s by institution and location separately in decreasing frequency order. The top-ranked values from the two sorted lists are taken as $BA(v)$.

The Girvan-Newman algorithm is chosen as our community detection algorithm because it does not hold bias against small-sized graphs. Since the detection algorithm is conducted on the v 's neighborhood graph, which is on comparatively small scale, the algorithms that hold bias to sparsely connected or small graphs are excluded from our consideration. On the other hand, the Girvan-Newman algorithm proceeds by removing the edges with the highest edge-betweenness [50] value iteratively, and the procedure is suitable to conduct on small-sized graphs.

As for the number of top values to take in step 4, it can be decided by the target user's number of friends and the frequency of sorted values. More friends indicate more experience, and more values should be taken. Meanwhile, the values whose frequency is comparable with that of the top one value could also be taken. Intuitively, the higher the frequency, the higher the probability the value is accurate.

2.4.2 Basic Attributes from Wall and Photos

The users in $U2$ display incomplete or no BA and conceal their friend lists from strangers, but some of their wall posts or photos are visible. We need to infer their BA s. Out of the dataset, 14.3% of the users belong to $U2$. It is formulated as:

$$U2 = \{v \mid v \in U \text{ and } PS(v) = (0, 0, x_1, x_2), x_1, x_2 = 0, 1 \text{ and } x_1 + x_2 > 0\};$$

Inference objective : $BA(v), v \in U2$;

Public information : v 's public wall posts or photos.

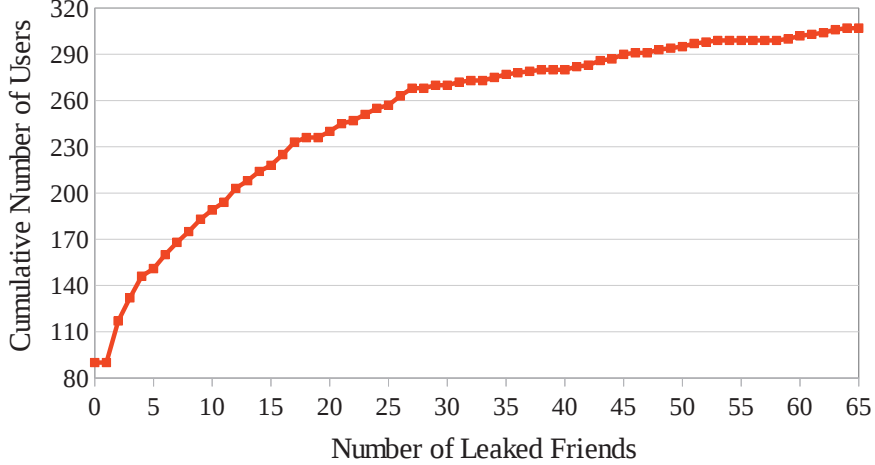


Figure 2.2: Number of Leaked Friends

Although the target user v 's friend list is private, a direct leakage of v 's connections is in v 's photos or wall posts where its friends leave comments or get tagged. Different numbers of connections are leaked for different users, depending on their activities and privacy settings on the wall and photo subpages. We randomly choose 330 users in the dataset seeds' neighborhood that belong to U_2 , and crawl their public photos and part of wall posts. The cumulative number of users having less than or equal to a certain number of leaked friends is depicted in Figure 2.2. While about 90 users have no friends leaked, over half of the users have more than five friends leaked and the maximum number of leaked friends is 295. If all the public wall posts are crawled, the number of leaked friends would increase.

Whereas v has some leaked friends, they may compose a small portion of v 's total friends. Namely, the leaked friends can be too sparse to form detectable communities in v 's neighborhood. Therefore, Scheme 1 is not applicable to users in U_2 . We seek to uncover $BA(v)$ in v 's leaked friends' neighborhood, instead of v 's neighborhood. First we traverse the directly leaked friends to retrieve their public friend lists and verify their connections to v . For those verified friends, their own friends can be traversed to obtain their neighborhood graphs and then detect communities in their neighbor-

hoods. As illustrated before, the *community feature* is supposed to be the most widely shared by community members. Here v is classified to a certain community in each of the verified friends' neighborhood, and it should have a high probability to share the *community feature*. Accordingly, the steps to reveal $BA(v)$ are detailed below as

Scheme 2:

1. Look through v 's wall and photos to retrieve leaked friends.
2. Traverse each leaked friend to retrieve its friend lists if public and verify its connection with v .
3. For each verified friend u , traverse its friends and detect communities in u 's neighborhood using the Girvan-Newman algorithm, resulting in $GC(1)$, $GC(2)$, \dots , $GC(n)$; if $v \in GC(k)$, find the corresponding *community feature* $A(k)$ and its frequency.
4. Merge and sort $A(k)$ s, found in v 's leaked friends' neighborhoods, in decreasing frequency order and identify $BA(v)$ in the top values.

Intuitively, the more friends leaked, the more *community features* can be found to increase the inference accuracy. Figure 2.2 demonstrates the possibilities of conducting the scheme. However, some users may display their photo and wall subpages but no comments are there; hence no friends are leaked. These cases are treated the same as these users in $U4$.

Besides, Scheme 2 could also be improved by assigning weights to the leaked friends, under the observation that those friends who comment or leave messages to user v might be closer to v than other friends. Higher priority could be given to the *community feature* found in those closer friends.

2.4.3 Friends from Wall and Photos

Those users who conceal friend lists but display some wall posts or photos are categorized into $U3$. We need to infer their FLs . As Table 2.1 shows, 15.4% of users belong to $U3$. The scenario is formulated as:

$$U3 = \{v \mid v \in U \text{ and } PS(v) = (x, 0, x_1, x_2), x, x_1, x_2 = 0, 1 \text{ and } x_1 + x_2 > 0\};$$

Inference objective : $FL(v), v \in U3$;

Public information : v 's public wall posts or photos.

We aim to uncover v 's full friend list while there are some directly leaked friends from v 's wall or photo subpages. Therefore, the inference task can be interpreted as traversing near v 's neighborhood graph starting from the leaked friends and ascertaining whether those reachable users are v 's friends. A few important issues must be considered to make the traversal practical. First, considering that the number of reachable users increases exponentially with the traversal depth, we should limit the depth so that the traversal is doable. Second, the v 's neighborhood graph may be disconnected; thus, if there are components with no starting friends inside, it is arduous to measure the distance between disconnected components in hops by traversing beyond v 's neighborhood. We use the word *component* to refer to a connected subgraph within v 's neighborhood. Third, for traversed users having private friend lists, it is difficult to distinguish whether they are v 's friends.

Taking these practical issues into account, we refrain the traversal from going beyond v 's neighborhood graph. The traversal can be conducted in a breadth-first manner, starting from the leaked friends as roots. It proceeds only on those users whose friend lists include v , and stops on users whose friend lists exclude v . Those traversed users with private friend lists could be gathered together for further verifi-

ation. Overall, the inference scheme consists of two steps and are detailed below as

Scheme 3:

1. Traverse the v 's neighborhood graph starting from the leaked friends as Algorithm 2.1 specified.
2. Determine the connectivity between v and traversed users who have private friend lists.

Algorithm 2.1 uses the following notations:

$R(v)$: the set of users that are yet to be traversed in the coming iteration;

R : the set of users that are to be traversed in the current iteration;

$T(v)$: the set of users that have been traversed;

$C(v)$: the set of users that have been traversed but have their friend lists private.

Initially, $R(v)$ consists of the leaked friends from photos and walls, while $T(v)$, $C(v)$, and $FL(v)$ are empty. Each iteration represents the traversal of users a certain depth away from roots. The algorithm terminates when no users traversed in the previous round are friends of v , that is $R(v)$ is empty. Furthermore, the algorithm could be adjusted to terminate in advance by confining the traversal depth. The depth can be recorded by counting the number of iterations, and the traversal terminates when the depth limit has been reached.

When the traversal algorithm terminates normally, all v 's friends who have public friend lists and are in the same components with the leaked friends should be included in the derived set $FL(v)$. But users who are in different components from the leaked friends cannot be reached. This limitation is due to the feasibility concerns of Scheme 3. However, as the evaluation result in Section 2.5.2 indicates, on average the largest

Algorithm 2.1 Constrained Breadth-first Traversal

Input: $R(v)$ = leaked friends**Output:** $FL(v), C(v)$

```
while  $|R(v)| > 0$  do
   $R = R(v)$ ;
   $R(v) = \{\}$ ;
  for  $u \in R$  do
    Retrieve  $FL(u)$ ;
     $T(v) = T(v) + \{u\}$ ;
    if  $FL(u)$  is private then
       $C(v) = C(v) + \{u\}$ ;
    else
      if  $v \in FL(u)$  then
         $FL(v) = FL(v) + \{u\}$ ;
        for  $w \in FL(u)$  do
          if  $w \in T(v)$  then
            pass;
          else
             $R(v) = R(v) + \{w\}$ ;
          end if
        end for
      end if
    end for
  end if
end for
end while
```

component in a user's neighborhood consists of over 75% of its friends. In other words, a leaked friend is likely to be included in the largest component; thus the majority of v 's friends are reachable from the leaked friends. Besides, as the component size and edge density vary in v 's neighborhood, the traversal complexity differs.

Complexity of Algorithm 2.1. The complexity of Algorithm 2.1 is analyzed in terms of the number of users whose information have to be retrieved. Assume that all users' numbers of friends are at the same magnitude, denoted as f . Algorithm 2.1 constrains the traversal to be within two hops away from the target user v ; and thus all v 's friends and its friends' friends are traversed in the worst case. We first take

the v 's f friends into count; and then we count its friends' friends as follows. In the algorithm, each user can only be traversed once. Thus, counting v 's friends' friends should exclude v 's friends. Let $G = (V, E), V = FL(v)$ denote v 's neighborhood graph; and then for each $u \in V$, $f - degree(u)$ of its friends would be counted, which excludes v 's friends. Thus, $\sum_{u \in V} f - degree(u)$ more users should be counted, that is, $f^2 - \sum_{u \in V} degree(u)$, in which $\sum_{u \in V} degree(u) = 2|E|$ according to graph theory. In total, the algorithm is in $\mathcal{O}(f + f^2 - 2|E|)$. Therefore, the more densely v 's friends connect to each other, the fewer users have to be traversed. The complexity varies between $\Theta(f^2)$ and $\Theta(f)$. The best case is when v 's friends compose a complete graph, i.e. $|E| = \frac{f(f-1)}{2}$, then the complexity is $\mathcal{O}(f)$. When the algorithm terminates by limiting the traversal depth, the complexity would be lower.

As for the second step of Scheme 3, i.e., distinguishing the connectivity between v and traversed users who have private friend lists, the traditional link prediction algorithms such as common friends or Katz [65] can be employed.

2.4.4 No Leaked Friends

The users holding the strictest privacy settings are categorized into $U4$. These users set friends, wall and photo subpages as private and display some or no profile information. The users in this category constitute about 22.4% of the dataset. We need to infer both their FLs and BAs . While the inference schemes presented before start from some friend connections, the users in $U4$ display none of their friends.

Other means have to be sought to identify possible friends. One source to seek is the special friends or family member sections. Otherwise, the search people function could be exploited by using a user's location or institution, if provided, as keywords. Then, the search results can be traversed one by one to check whether the target

user is included in their friend lists. As long as one of the target user’s friends with public friend lists can be found, previous schemes can also be conducted to reveal its *targetProfile*. Otherwise, their privacy can not be inferred by our schemes.

In the next section, we apply these schemes to the dataset presented in Section 2.3 to quantify the privacy that can be compromised in each case.

2.5 Evaluation

The *BA* inference schemes are conducted on users who display their *BA* values, and the *FL* inference schemes are conducted on users who display their *FL* values; otherwise, the ground truth is not available for verification.

For the *targetProfile* inference, evaluation bias may be induced in the results when a user’s public profile is incomplete or fallacious. Considering the real name policy of Facebook [6], the problem of profile authenticity will not be as significant as incompleteness, which results in false positives. Especially for the location attribute values, only hometown and current city are available in the ground truth, while schemes 1 and 2 can also infer other cities where a user has ever stayed, such as those associated with the institutions where the user has ever been. Hence, the actual location inference accuracy should be higher than what the results illustrate.

2.5.1 Inferring Basic Attribute Values

Scheme 1 is evaluated first, which can be applied to the users with public friend lists. Out of the dataset, there are 909 users all of whose friends are in the dataset; thus, scheme 1 is applied to those users, referred to as evaluated users. Those who display nothing in their profiles are excluded due to the lack of ground truth for verification. Besides, users with more than 1,000 friends are excluded from the evaluation results.

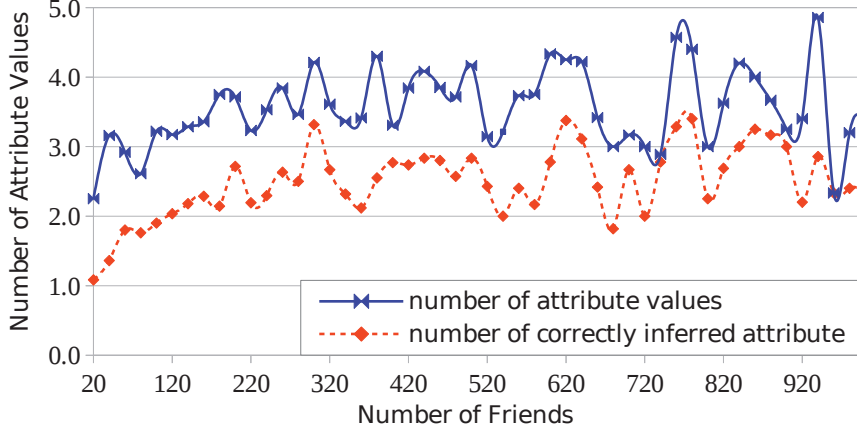


Figure 2.3: Number of Correctly Inferred Attribute Values

They consist of 5.17% of the total evaluated users, but less than three, if not zero, users fall into each user sample bin in this range; sparsity of user sample isn't likely to result in representative evaluation result.

We use the “igraph” [12] library to detect communities in each evaluated user's neighborhood with the Girvan-Newman algorithm [50]. In each community, the most frequently shared basic attribute value, the *community feature*, can be either a location or an institution value. We identify both the most-shared institution and location values when the community size is above average, and the one with lower frequency is called the *additional feature* of the community. Then we merge and sort those community features and additional features separately in decreasing frequency order by location and institution, respectively. The top ranked values are taken as the user's inferred basic attribute values.

We evaluate the basic attribute inference schemes from the following three aspects. (1) How many basic attribute values could be inferred? The number of public attribute values in evaluated users' homepages which are taken as ground truth, varies from user to user; thus, the number of correctly inferred basic attribute values for each user should be measured. (2) How accurate are inferred values? The number of top

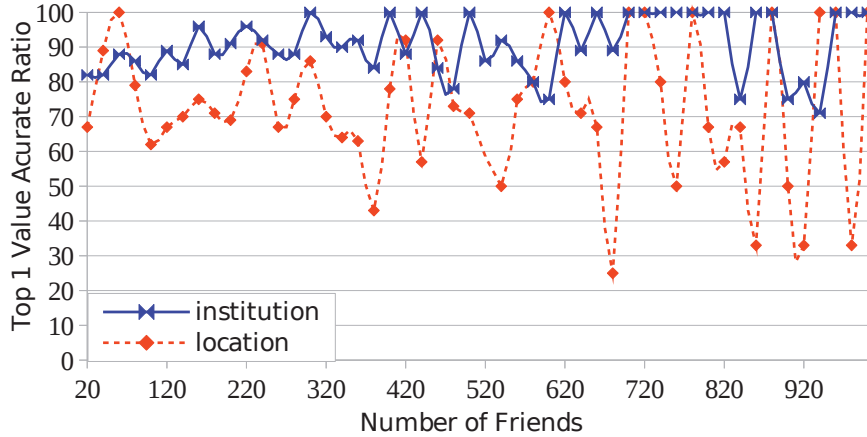


Figure 2.4: Inference Accuracy

values from sorted community features, taken as inferred basic attribute values, can be adjusted; hence the accuracy of each value in the top rank should be measured. (3) Whether the number of correctly inferred basic attribute values and the inference accuracy are affected by the number of the evaluated user’s friends. Since the basic attribute values are inferred from the target user’s friends’ information, we want to know whether the number of friends affects the inference accuracy or number. Figures 2.3 to 2.6 give answers to those questions one by one. In all these figures except for Figure 2.6, the x-axis value is the number of users’ friends and the y-axis value is the average value of users whose number of friends fall into the 20 user sample bin.

Figure 2.3 depicts the number of correctly inferred basic attribute values compared to the number of basic attribute values in ground truth. The figure shows that more attribute values could be inferred for users with more than 100 friends compared to those with less friends. It verifies the previous claim that the more friends a users has, the more attribute values could be derived; but the differences among users who have more than 120 friends are not significant. On average, more than two attribute values could be correctly inferred. Attribute values that are not reflected in a user’s community features cannot be inferred; one possible reason is that the user is not

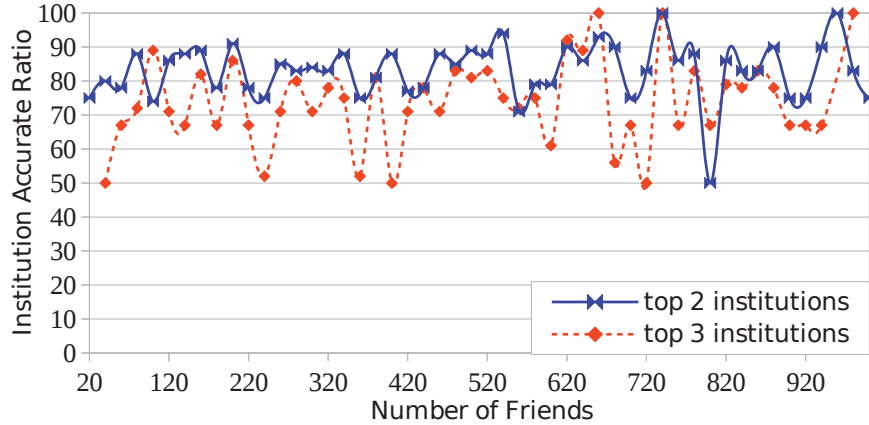


Figure 2.5: Top Institutions Accuracy

active in certain OSN communities, or its residence in a certain institution or city is too short to form a community.

The accuracy of the top values taken as inferred basic attribute values are shown in Figures 2.4 and 2.5. The accurate ratio is defined as the ratio between the number of verified inferred attribute values and the number of inferred values. In Figure 2.4 top 1 institution and location are taken as inferred values while in Figure 2.5 top 2 and top 3 institutions are taken as inferred values.

Figure 2.4 shows that the inference accurate ratio for institution is about 90% on average, and overall, the more friends the target user has, the higher the average accurate ratio is. Meanwhile the accurate ratio of location is not as good due to the false positives incurred by the incomplete ground truth of location values. As we mentioned at the beginning of this section, only hometown and current city are included in the ground truth for location while we infer all the places that the user has ever been. In addition, the accurate ratio of the top 1 location value for users with more than 500 friends fluctuates more strongly. One reason is that usually the larger the number of friends, the more experience a user has or the more locations a user has ever been, and in turn the less chance for the hometown or current city to be

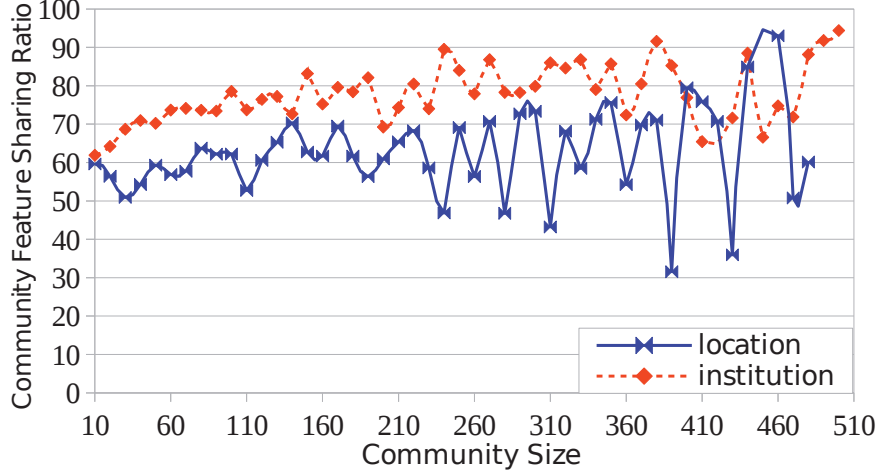


Figure 2.6: Community Feature Sharing

derived as the top 1 inferred location value. Another reason is that users with more than 500 friends are sparse at some point compared to users with fewer friends; thus the accurate ratio cannot be averaged and tends to go extremes due to the sparse user sample. This also explains the higher variance for those users in Figures 2.3 and 2.5.

Though the missing of ground truth for location leads to false positives, each institution is usually associated with a location; as long as institutions are correctly inferred, corresponding locations could be derived. Hence, we further evaluate the accurate ratio of inferred institution information in Figure 2.5. Figure 2.5 depicts the accurate ratio of top 2 and top 3 ranked institution values. It shows the accuracy of top 2 institution values is over 80%, which on average is higher than that of top 3 institution values. It verifies our claim that higher-ranked community features hold higher probability to be shared by the target user. Besides, the accurate ratio is not largely affected by the number of users' friends.

For users belonging to U_2 , we first measure the community feature sharing ratio to evaluate their basic attribute values inference accuracy, since their basic attributes are derived from the community feature in their leaked friends' neighborhood. Figure 2.6

depicts the community feature sharing ratio, and x-axis value is the community size. More than 8,500 communities are detected in the evaluated users' neighborhood. On average, the sharing ratio is higher when the community feature is an institution value compared to when it is a location value. This difference can also be explained by the ground truth incompleteness of location information. Though the community features are not 100% shared by all members, they will not be directly taken as the inferred basic attribute values and the wrong community features will be eliminated in the later steps of Scheme 2.

We further evaluate the inference accuracy of Scheme 2 on some of the dataset's seed users which belong to $U2$. Because seed users are from the same institution and location, their information ground truth scraped from users' homepages are complemented by that fact. We detect those seed users' community memberships in their friends' neighborhood, and take the top ranked community features as their inferred attribute values. As a result, the inference accuracy of top 1 ranked feature is 100%.

In summary, for users who conceal their basic attribute values but have their friend list public or some friends leaked from other profile sections, those value could be uncovered with high accuracy by exploiting their friends' information.

2.5.2 Inferring Friend List

For a user v in $U3$, v 's retrievable friends, according to Scheme 3, are confined to those who are in the same components with the leaked friends. As defined in Section 2.4.3, a component is a connected subgraph within v 's neighborhood. We first measure the components in users' neighborhoods. Out of the evaluated users, most of their neighborhood graphs are disconnected, on average 20 components exist and the number of components increases with the number of friends. While there are

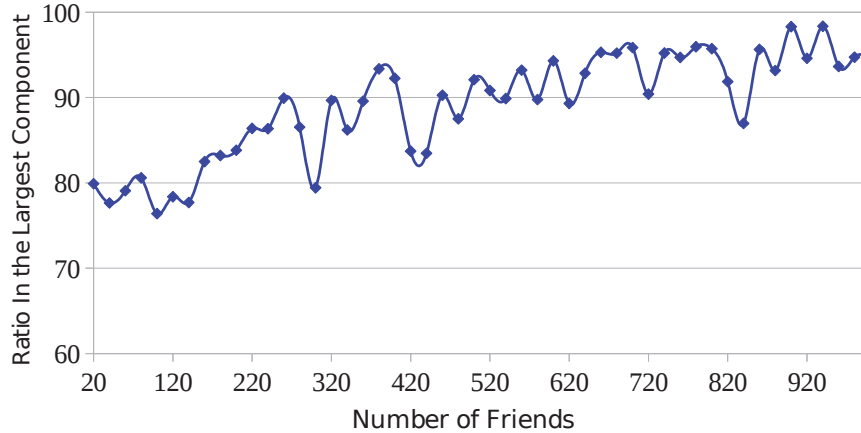


Figure 2.7: Friends in the Largest Component

a noticeable number of components, most of them are small. Figure 2.7 illustrates the ratio of a user’s friends that are in the largest component, over 85% of friends on average are included in the largest component. The more friends a user has, the larger portion of friends are in the largest component. As the leaked friends are likely to be in the largest component, a majority of friends could be reached from them.

In Figure 2.8, the ratio of traversed friends in the evaluated users’ neighborhoods is illustrated, and the traversal starts from different number of roots in one hop away. Each curve represents a different number of roots, which are randomly chosen from target user’s friends. For users with fewer than 100 friends, a majority of friends could be traversed in one hop from five roots, while for users with more friends, about 10%, 25%, and 35% of friends could be traversed in one hop away from two, five, and ten roots, respectively. Over all, the more friends a user has, the more of its friends can be reached via traversal given the same number of roots and hops.

Figure 2.9 indicates the ratio of friends traversed in two hops away. About 70% of friends could be traversed from 5 roots, and 80% of friends could be traversed from 10 roots. The curve for two roots fluctuates more violently because the choice of roots affects the traversal path and a high-degree node results in more retrieved friends.

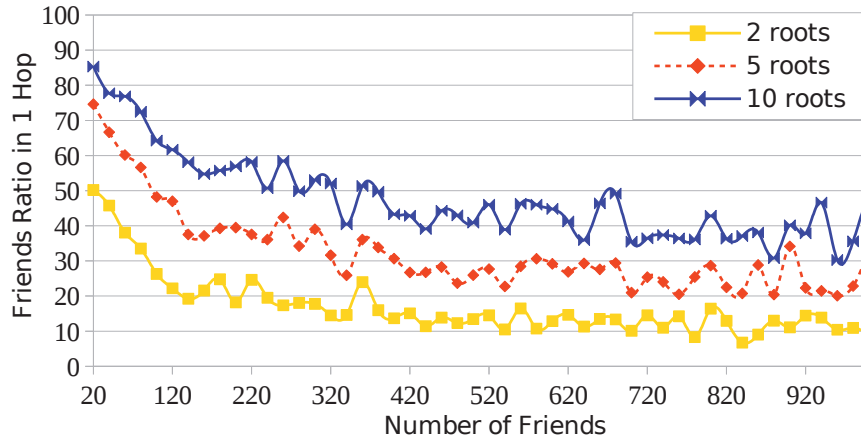


Figure 2.8: Traversed Friends Ratio in 1 Hop

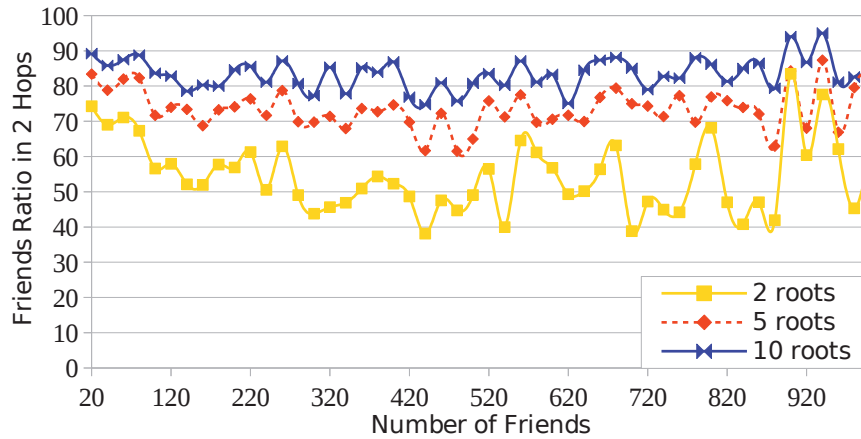


Figure 2.9: Traversed Friends Ratio in 2 Hops

When starting from 5 or 10 roots, the high-degree nodes stand a higher chance to be traversed as roots or within two hops. Still, on average about half of a user’s friends could be retrieved from two randomly chosen roots in two hops. Interestingly, the ratio is not clearly affected by users’ number of friends. It means that no matter how many friends a user has, most of its friends are closely connected while some are estranged from others.

To sum up, for users who conceal their friend lists but display other profile sections from which some of their friends could be leaked, our algorithm is able to recover over half of their friends. in two hops. The complexity of the traversal algorithm ensures

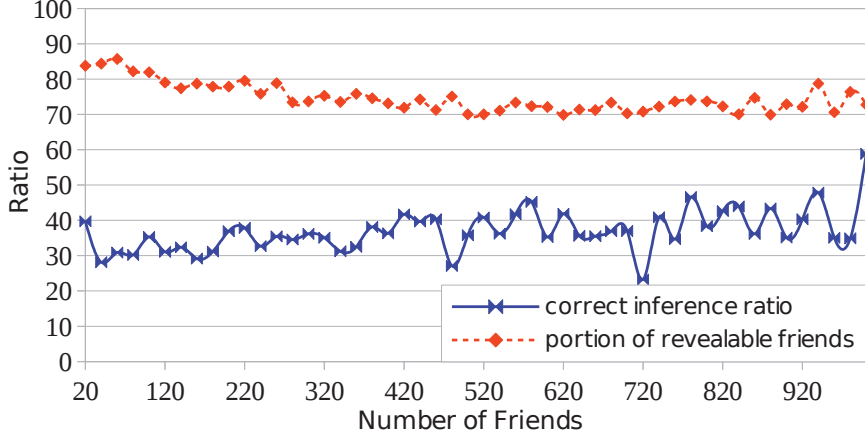


Figure 2.10: Private-Friends Inference Ratio

the traversal can be conducted in limited resource.

After that, we measure the second step of scheme 3, i.e., to distinguish the connections between user v and the traversed users who have private friend lists. Those users are those who connected to v 's friends and have private friend lists. The number of common friends is taken as the metric to infer the connections. Those private-friend-listed users are sorted by their numbers of friends shared with v , which is leaked from v 's public-friend-listed friends. The top quarter of users are taken as v 's hidden friends. Figure 2.10 illustrates the inference accuracy, and it also illustrates the total revealable friends ratio, which consists of both the public-listed friends and those hidden friends. Compared to the results of [65] which also used common neighbors as the metric to infer co-authorship, our accuracy is slightly higher. In total, for users belonging to $U3$, more than 70% of their friends could be correctly revealed on average by Scheme 3.

Users in $U4$ hide all connections, which is hardest to infer their *targetProfile*. However, if some of their friends are known beforehand or can be found by using the search people function mentioned in Section 2.4.4, their *targetProfile* can be inferred and evaluated similar as stated above.

2.6 Discussion

While our approach explores a user’s information visibility from the perspective of a stranger, it cannot know the privacy customization to the user’s friends. However, the privacy setting for strangers can only be stricter than that for friends. In other words, friends must be able to access more information than strangers. Thus, if some private information could be correctly inferred by a stranger, the inference can also be reproduced by friends.

If a user does not post certain profile item on Facebook such as education, we cannot know whether the invisibility is due to privacy setting or vacancy. However, if the inferred information could be verified based on the ground truth retrieved from other sources, we still view such a case as privacy leakage.

In order to build the ground truth, our experimental data only samples users who display their full profiles to strangers. However, we speculate that other users with more strict privacy settings are also inclined to be more prudent in setting up connections. And since their friend circles are created in a more moderated manner, community feature detection and neighborhood graph traversal should perform no worse. Therefore, our evaluation presents a plausible privacy breach of average users.

The profile inference schemes proposed in this chapter are not limited to Facebook. They could also be applied to other OSNs that enable privacy configuration and allow users to post a variety of data other than profile and connection. Those OSNs include MySpace, Google+, and Renren, in which users could also upload photos, leave messages or comments, and customize the visibility of different types of information. When the accessibility of a user’s profile or connections is constrained, the information revelation could be initiated from public connections in the friend list or posts from friends by using our schemes 1, 2 or 3.

2.7 Summary

In this chapter, we investigated the unintentional privacy disclosure of OSN users even with the protection of privacy settings. We first examined users' privacy settings on different information sections of a large dataset collected from Facebook. Then, for each possible privacy configuration, we proposed corresponding schemes to reveal basic profile and connection information starting from leaked public connections on the target user's OSN homepage. Finally, using our dataset, we quantified the achievable privacy exposure in each case, and measured the accuracy of our privacy inference schemes given a different amount of public information. The evaluation results indicate that a user's private basic profile could be inferred with high accuracy, while a user's covert connections could be uncovered in a significant portion based on even a small number of directly leaked connections.

Our privacy inference schemes can be conducted by attackers without much resources; and those schemes are applicable to users adopting specific privacy settings. The dataset statistics show that a majority of users are among that group. Therefore, the privacy of those users could be undermined facilely and the actual information privacy level of them may fail to meet what their privacy configuration specifies. We discussed that our privacy inference schemes could be applied to other OSNs that provide similar features as Facebook.

Profiling Social Behaviors for Compromised Account Detection

Account compromise is a serious threat to users of Online Social Networks (OSNs). While relentless spammers exploit the established trust relationships between account owners and their friends to efficiently spread malicious spam, timely detection of compromised accounts are quite challenging due to the well established trust relationship between the service providers, account owners, and their friends. In this chapter, we study the *social behaviors* of OSN users, i.e. their usage of OSN services, and the application of which in detecting compromised accounts. In particular, we propose a set of social behavioral features that can effectively characterize the user social activities on OSNs. We validate the efficacy of these behavioral features by collecting and analyzing real user clickstreams to an OSN website. Based on our measurement study, we devise individual user's social behavioral profile by combining its respective behavioral feature metrics. A social behavioral profile accurately reflects a user's OSN activity patterns. While an authentic owner conforms to its account's social behavioral profile involuntarily, it is hard and costly for impostors to feign. We evaluate the capability of the social behavioral profiles in distinguishing different OSN

users, and our experimental results show the social behavioral profiles can accurately differentiate individual OSN users and detect compromised accounts.

3.1 Motivation

Compromised accounts in Online Social Networks (OSNs) are more favorable than sybil accounts to spammers and other malicious OSN attackers. Malicious parties exploit the well-established connections and trust relationships between the legitimate account owners and their friends, and efficiently distribute spam ads, phishing links, or malware, while avoiding blocking by the service providers. Offline analysis of tweets and Facebook posts [48, 52] reveal that most spam are distributed via compromised accounts, instead of dedicated spam accounts. Recent account hacking incidents [1, 2] in large scale in popular OSNs further evident this trend.

Unlike dedicated spam or sybil accounts, which are created solely to serve malicious purposes, compromised accounts are originally possessed by benign users, and later hijacked by cyber criminals. While dedicated malicious accounts can be simply banned upon detection, compromised accounts cannot be handled likewise due to the negative impact to normal user experiences (e.g., those accounts may still be actively used by their legitimate owners). Major OSNs today employ IP geolocation logging to battle against account compromise [3, 9]. However, this approach is known to suffer from low detection granularity and high false positive rate.

Previous research on spamming account detection [47, 48, 52, 105] mostly cannot distinguish compromised accounts from sybil accounts, with only one recent study by Egele *et al.* [44] features compromised accounts detection. Existing approaches involve account profile analysis [94, 105] and message content analysis [44, 47, 52, 92] (e.g., embedded URL analysis [52, 92] and message clustering [44, 47]). However, account

profile analysis is hardly applicable for detecting compromised accounts, because their profiles are the original common users' information which is likely to remain intact by spammers. URL blacklisting has the challenge of timely maintenance and update, and message clustering introduces significant overhead when subjected to a large number of real-time messages.

Instead of analyzing user profile contents or message contents, we seek to uncover the behavioral anomaly of compromised accounts by using their legitimate owners' history social activity patterns, which can be observed in a lightweight manner. To better serve users' various social communication needs, OSNs provide a great variety of online features for their users to engage in, such as building connections, sending messages, uploading photos, browsing friends' latest updates, etc. However, how a user involves in each activity is completely driven by personal interests and social habits. As a result, the interaction patterns with a number of OSN activities tend to be divergent across a large set of users. While a user tends to conform to its social patterns, a hacker of the user account who knows little about the user's behavior habit is likely to diverge from the patterns.

Therefore, as long as an authentic user's social patterns are recorded, checking the compliance of the account's upcoming behaviors with the authentic patterns can detect account compromise. Even though a user's credential is hacked, a malicious party cannot easily obtain the user's social behavior patterns without the control of the physical machines or the clickstreams. Moreover, considering that for a spammer, who carries very different social interests from those of regular users (e.g., mass spam distribution vs. entertaining with friends), it is very costly to mimic different individual user's social interaction patterns, as it will significantly reduce spamming efficiency.

In sight of the above intuition and reasoning, we first conduct a study on online

user social behaviors by collecting and analyzing user clickstreams [35, 86, 97] of a well known OSN website. Based on our observation of user interaction with different OSN services, we propose several new behavioral features that can effectively quantify user differences in online social activities. For each behavioral feature, we deduce a behavioral metric by obtaining a statistical distribution of the value ranges, observed from each user’s clickstreams. Moreover, we combine the respective behavioral metrics of each user into a social behavioral profile, which represents a user’s social behavior patterns.

To validate the effectiveness of social behavioral profile in detecting account activity anomaly, we apply the social behavioral profile of each user to differentiate clickstreams of its respective user from all other users. We conduct multiple cross-validation experiments, each with varying amount of input data for building social behavioral profiles. Our evaluation results show that social behavioral profile can effectively differentiate individual OSN users with accuracy up to 98.6%, and the more active a user, the more accurate the detection.

3.2 Related Work

Schneider et al. [86] and Benevenuto et al. [35] measured OSN users’ behaviors based on network traffic collected from ISP. Both work analyzed the popularity of OSN services, session length distributions, and user click sequences among OSN services, and they discover that browsing accounts for a majority of users’ activities. Benevenuto et al. [35] further explored user interactions with friends and other users multiple hops away. While their work primarily emphasize the overall user OSN service usage, and aim to uncover general knowledge on how OSNs are used, this chapter studies users’ social behavior characteristics for a very different purpose. We investigate the

characterization of individual user’s social behaviors and targets differentiating account usage anomaly. More over, we propose several new user behavioral features and perform measurement study at a fine granularity.

While most previous research on malicious account detection cannot differentiate compromised accounts from spam accounts, Egele et al. [44] specifically studied the detection of compromised accounts. By recording a user’s message posting features, such as timing, topics and correlation with friends, they detected irregular posting behaviors; on the other hand, all messages in a certain duration are clustered based on the content and the clusters in which most messages are posted by irregular behaviors are classified as from compromised accounts. While they also leverage certain user behavior feature to discern abnormality, we use a different and more complete set of metrics to characterize users’ general online social behaviors, instead of solely focusing on message posting behavior. Additionally, our technique do not rely on deep inspection and classification of message contents, therefore it is scalable for large social networks.

Wang et al. [97] proposed sybil accounts detection via analyzing clickstreams. They differentiated sybil and common users’ clicks, in terms of interarrival time and click sequence, and found that considering both factors leads to better detection results. Since sybils are specialized fake identities owned by attackers, their clickstream patterns significantly differ from normal users. However, for compromised accounts, their clickstreams can be a mix from normal users and spammers, As a result, methods in [97] cannot handle compromised accounts well. This chapter aims to uncover users’ social behavior patterns and habits from the clickstreams, with which we can perform more accurate and delicate detection on behavioral deviation.

Regarding spammer detection, [90] and [61] set up honeypot accounts to harvest spam and identify common features among spammers, such as URL ratio in their

messages and friends choice; using those features, both employ classification algorithms to detect spammers. Yang et al. [105] introduced new features of spammers involving with their connection characteristics to achieve better accuracy. Thomas et al. [94] analyzed the features of fraudulent accounts bought from the underground market and developed a classifier using the features to retrospectively detect fraudulent accounts. Instead of focusing on malicious accounts, Xie et al. [103] proposed to vouch normal users based on the connectivities and interactions among them.

As for spam detection, Gao et al. [47] proposed a real-time spam detection system, which consists of a cluster recognition system to cluster messages and a spam classifier using six spam message features. Thomas et al. [92] thrived to detect spam by identifying malicious URLs in the message content. In [48, 52], the authors conducted offline analysis to characterize social spam in Facebook and Twitter, respectively. They found that a significant portion of spam are from compromised accounts instead of spam accounts. Meanwhile, Yang et al. [104] investigated connectivities among identified spammers. Other malicious account detections exploit the differences on profile or connectivity information between normal and malicious accounts [37, 88, 96].

Users' social behavior analysis has also been applied for other purposes. Wilson et al. [100] analyzed user interactions with friends from the trace in Facebook profiles to improve performance for sybil detection while reducing its complexity. In [29, 85], the authors correlated users' personalities with their OSN service usages.

3.3 User Social Behaviors Study

In this section we first propose several social behavior features on OSNs, and describe in detail how they can reflect user social interaction differences. Then, we present a measurement study of user behavior diversity on our proposed features by analyz-

ing real user clickstreams of a well known OSN, the Facebook, with respect to our proposed features.

3.3.1 Social Behavior Features

We categorize user social behaviors on an OSN into two classes, extroversive behaviors and introversive behaviors. Extroversive behaviors, such as uploading photos and sending messages, result in visible imprints to one or more peer users; introversive behaviors, such as browsing other users' profiles and searching in message inbox, however, do not produce observable effects to other users. While most previous research only focus on the extroversive behaviors, such as public posting [44], we study both classes of behaviors for a more complete understanding and characterization of user social behaviors.

3.3.1.1 Extroversive Behavior Features

Extroversive Behaviors directly reflect how a user interacts with its friends online, and thus they are important for characterizing a user's social behaviors. We decompose extroversive behaviors into the following four major aspects.

- *First Activity:*

The first extroversive activity a user engages in after logging in an OSN session can be habitual. Some users often start from commenting on friends' new updates; while some others are more inclined to update their own status first. The *first activity* feature aims to capture a user's habitual action at the beginning of each OSN session.

- *Activity Preference:*

How often a user engages in each type of extroversive activities relates to their per-

sonalities [29]. Some users like to post photos, while some others spend more time responding to friends' posts; some mostly chat with friends via private messages, while some others always communicate by posting on each other's public message boards. Typical OSNs provide a great variety of social activities to satisfy their users' communication needs, for example, commenting, updating status, posting notes, sending messages, sharing posts, inviting others to an event, etc. As a result, this feature can provide a detailed portrayal of a user's social communication preferences.

◦ *Activity Sequence:*

The relative order a user completes multiple extroversive activities. While users have their preferences on different social activities, they may also have habitual patterns when switch from one activity to another. For instance, after commenting on friends' updates, some users often update their own status, while some other users prefer to send messages to or chat with friends instead. Therefore, the action sequence feature reflects a different social behavioral pattern from the *activity preference*.

◦ *Action Latency:*

The speed of actions when a user engages in certain extroversive activities reflects the user's social interaction style. Many activities on OSNs require multiple steps to complete. For example, posting photos involves loading the upload page, selecting one or more photos, uploading, editing (e.g., clipping, decorating, tagging, etc.), previewing and confirmation. The time a user takes to complete each action of a given activity is heavily influenced by the user's social characteristics (e.g., serious vs. casual) and familiarity with the respective activity; but it doesn't directly reflect how fast a user acts due to different content complexity. The *action latency* feature is proposed to provide more fine-grained and accurate metric.

3.3.1.2 Introversive Behavior Features

Although invisible to peer users, introversive behaviors make up the majority of a user’s OSN activity; as studied in previous work [35, 86], the dominant (i.e. over 90%) user behavior on an OSN is browsing. Through introversive activities users gather and consume social information, which helps them to form ideas and opinions, and eventually, establish social connections and initiate future social communications. Hence, introversive behavior patterns make up an essential part of a user’s online social behavioral characteristics. We propose the following four features to portray a user’s introversive behavior.

- *Browsing Preference:*

The frequency a user visits various OSN page types depicts its social information preferences. Typical OSNs classify social information into different page types. For instance, profile pages contain personal information of the account owners, i.e., names, photos, interests etc.; the homepage compose of the account owner’s friends’ latest updates while a group page consists posts or photos shared by group members. Users’ preferences on various types of social information naturally differ by their own interests, and the *browsing preference* feature intends to reflect this difference by observing users’ subjective behaviors.

- *Visit Duration:*

The time a user spends on visiting each webpage depicts another aspect of its social information consumption. Intuitively, users tend to spend less time on information that are “good-to-know”, while allocate more time on consuming information that are “important”, and their judgments are made based on their own personal interests. For example, some users prefer to stay on their own homepage reading friends’ comments and updates, while some others tend to spend more time reading others’

profile pages. The *visit duration* feature aims at capturing the social information consumption patterns for different users.

○ *Request Latency:*

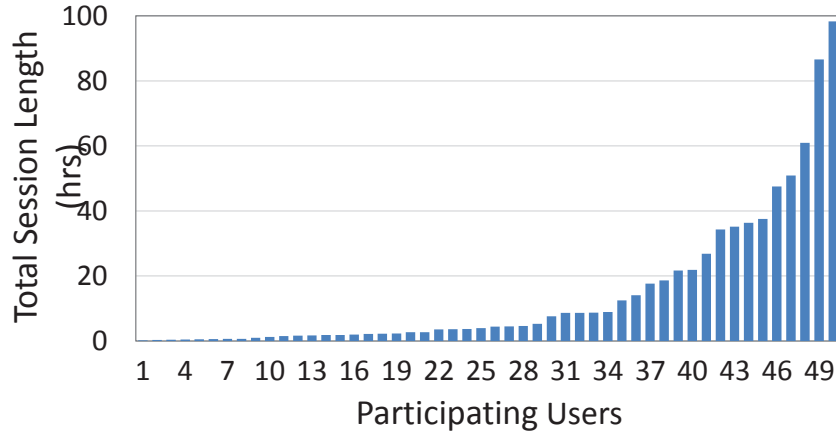
During a single visit to a webpage, a user may request multiple pieces of information. For example, browsing through a photo album requires loading each photo inside the album; reading comments from friends may also require “flipping” through many “pages” because only a limited amount of entries can be displayed at a time. Similar to the *action latency* feature for extroversive activities, the *request latency* feature provides fine-grained characterization of users’ social information consumption patterns.

○ *Browsing Sequence:*

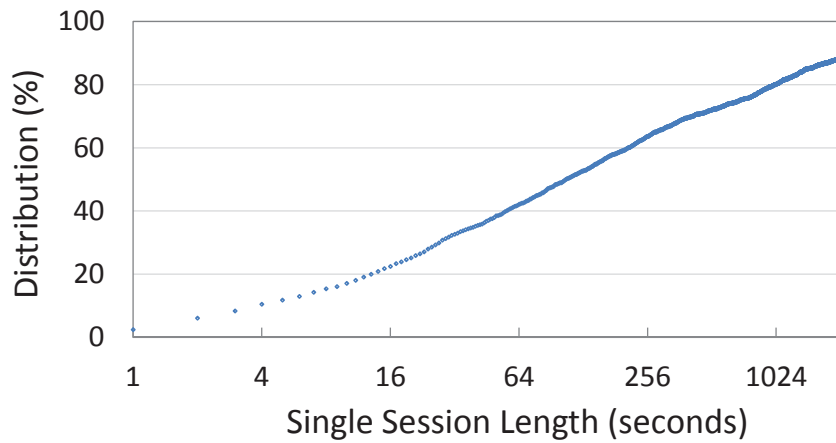
The order a user switches between different webpages reflects a user’s navigation patterns amongst different types of social information. OSNs usually provide easy navigation for users to move around various pages; a user on a friend profile page can directly navigate to another’s profile page, or go back to the homepage and then go to another friend’s profile page. How each user navigates during browsing can be habitual, and this feature intends to capture this characteristics.

3.3.2 Facebook Measurement Study

We conduct a measurement study of Facebook users to understand their online social behaviors. In order to observe both extroversive and introversive behaviors from the participating users, we collect information from the network perspective—we develop a browser extension to record user activities on Facebook in the form of clickstreams. In the following, we first present our data collection design and techniques, and an overview of the data set. Then, we detail the measurement results of user behavioral



(a) Cumulative Session Lengths



(b) Single Session Length Distribution

Figure 3.1: Facebook Data Set Overview Statistics

features.

3.3.2.1 Dataset Overview

We have recruited a total of 50 Facebook users for our measurement study—22 are graduate students at universities and the rest are recruited via Amazon Mechanical Turk or Odesk, both of which are popular online crowdsourcing marketplaces. For each user, we collect approximately three weeks of their Facebook activities. To ensure that the recruited users are actually normal Facebook users, we use their first

week as “trial” periods, during which we conduct manual review on the collected activity data.

The clickstreams in our dataset are organized in units of “sessions”. We denote the start of a session when a user starts to visit Facebook in any window or tab of a browser; the end of a session is denoted when the user closes all windows or tabs that visit Facebook, or navigates away from Facebook in all windows or tabs of the browser. Clickstreams from concurrently opened tabs/windows are grouped into a single session, but are recorded individually (i.e., events from one window/tab are not merged with those from another window/tab). In total, we have collected 2678 sessions.

We further process each clickstream before starting detailed measurements. By analyzing the request timestamp and URLs we detect and remove clickstreams in the “idle” periods—significantly long time intervals in which no user activity is observed. For example, users may go away from their computer while leaving their browser running. With idle periods removed, we plot the “effective” cumulative clickstream lengths for each participating users in Figure 3.1(a). We observe in this figure that the clickstream lengths follow exponential distribution. During a three week period, the least active user only accumulates half an hour of activities, while the most active user spends more than 80 hours on Facebook. We also plot the CDF of single session lengths across all users in Figure 3.1(b). It is evident that the distribution of single session length is heavy-tailed. While over 66% of sessions are within 300 seconds, more than 11% of sessions are over 2000 seconds long.

3.3.2.2 Feature Measurements

We first conduct a systematic study of services and webpages on Facebook. Based on request URL, we categorize 29 different types of extroversive activities that can

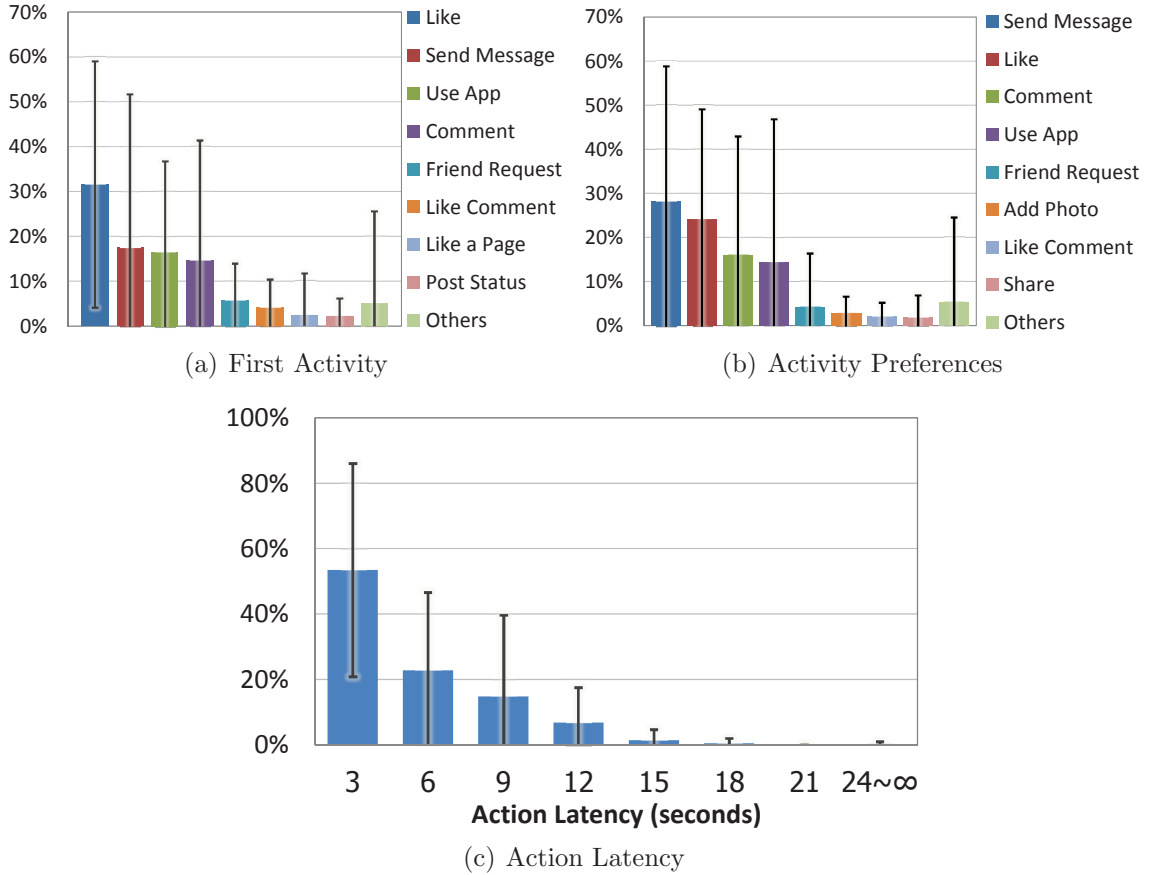


Figure 3.2: Combined Distributions of Extroversive Features

be used to interact with peer users; we also classify 9 types of Facebook webpages containing different kinds of social information, which users can browse privately (i.e. the introversive activities). With the mapping between the clickstream information and the user behaviors, we analyze each user’s clickstreams to extract their respective behavior patterns. We present the combined measurement results of each behavior feature for all users to show their value space, and finally we use an example to illustrate user behavior diversities.

Figure 3.2(a) shows the distribution of *first activity* in users’ OSN sessions. From this figure, we can observe that there are four most favorite activities, “like”, “send message”, “use app”, and “comment”, which account for about 80% of all 29 activ-

ities. The rest 25 types of activities are much less frequently engaged in by users, and account for about 20% of all activities. While the “send message”, “use app”, and “comment” activities have comparable proportions, the “like” activity has an obviously higher preference than the other three. The extraordinarily tall error bars indicate that users have significant diversities over all types of activities.

Figure 3.2(b) shows the distribution of extroversive activities users involve in, i.e., the *activity preference*. The primary features of this figure are similar to those of Figure 3.2(a): the same four activities dominate the distribution, and the user diversities on all activities are also very significant. However, there are some minor but observable differences. First, the relative orders of the four most favorite activities are different from Figure 3.2(a); in addition, the relative proportions of the four activities are more comparable. Second, the “add photo” and “share” emerge as “significant” activities (having a $> 2\%$ share of the total), while the “Like a page” and “post status” activities become “insignificant”. These differences indicate that user behaviors do vary significantly across different behavioral contexts.

We discern the user *action latency* by first grouping clickstreams belonging to individual user activities, and then measuring the interarrival time of consecutive HTTP requests within each group of clickstreams. The distribution of action latencies is presented in Figure 3.2(c), from which we can observe that users generally interacts with services at a fast pace. over 90% of actions have less than 7 second delay in-between. Action latencies in the 0 to 9 second range are diverse between individual users, resulting in tall error bars.

The user *browsing preferences* distribution is shown in Figure 3.3(a). We can see that visits to the “homepage” and “profile” account for 86% of all user browsing actions. Similar to Figure 3.2(a) and 3.2(b), large user diversities manifest on all

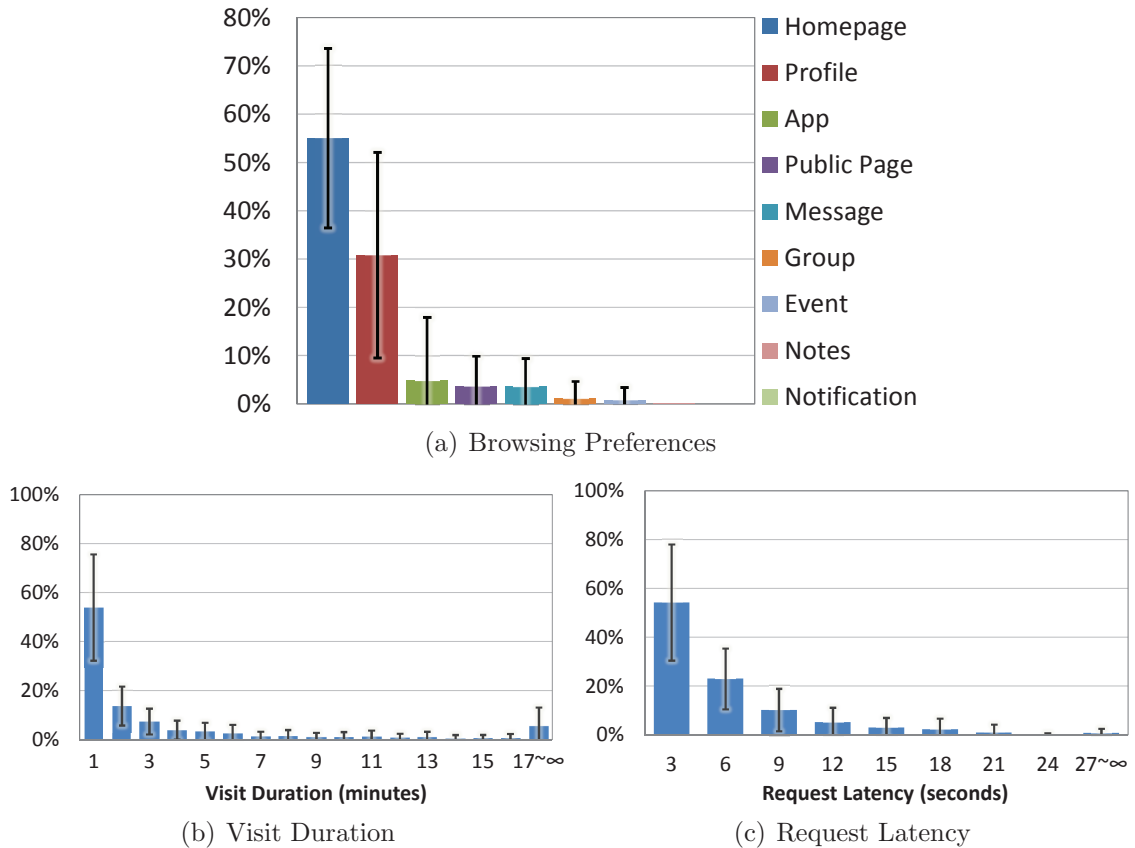


Figure 3.3: Combined Distributions of Introversive Features

types of webpages¹.

Figure 3.3(b) presents the distribution of webpage visit duration. With a heavy-tailed distribution, over 90% of visits last less than 600 seconds. Users tend to have highly divergent behaviors for visit durations in the 0 to 3 minute range. We study the *request latency* using the same technique for measuring *action latency*, and observe similar results. Shown in Figure 3.3(c), with 90% of inter-request delays less than 10 seconds, the latencies for request sending during webpage browsing are generally slightly larger than that for engaging in extroversive activities. User divergence is most obvious in the 0 to 9 second range.

¹We did not observe enough “notes” and “notification” page visits to derive meaningful statistics, so their data are considered as *unknown*.

Table 3.1: Feature Value Comparison

		User A	User B
Top First Activity		Click “like” button	“Like” a page
Top Activity		Post photo	“Like“ a page
Top Activity Trans.		Msg. → Msg.	“Like” page → “Like” page
Avg. Action Latency		4.36s/req.	2.31s/req.
Top Webpage		Profile	Homepage
Avg. Duration	Homepage	185.56s	175s
	Profile	134s	118s
Avg. Latency	Homepage	3.8s/req.	6.96s/req.
	Profile	4.13s/req.	4.75s/req.
Top Webpage Trans.		Profile → Profile	Homepage → Public page

We further study the *action latency* and *request latency* using the burstiness parameter [51]. The burstiness parameter has a value range of $[-1, 1]$, with 1 denoting complete randomness, and lower values signifying more regular behaviors. We find that, for *action latency*, the average value of burstiness parameters is -0.12, and 82% of the users’ burstiness parameters are negative, indicating that individual users tend to have regular action patterns; for *request latency*, the average burstiness parameter value is 0.035, which indicates slightly more randomness than *action latency*. In addition, 74% of users have lower burstiness parameter values for profile browsing than those for homepage browsing, indicating user browsing speed tends to be more random on homepage than on profile.

The *activity sequence* and *browsing sequence* consist of the distributions of usage for all pair-wise combinations of extroversive activities and webpages, respectively. Due to the large number of activities and webpages, the possible value spaces for these two features are very large. Normal user activities tend to explore only a small portion of these feature value spaces. We observe that, on average, each user’s extroversive activities only include 9.3 out of 841 (1.1%) possible activity combinations, and each user’s webpage visits only include 7.3 out of 81 (9%) possible webpage combinations.

Our measurement study shows that we can discern user online social behavior characteristics by analyzing their clickstreams. The results confirm that given a large number of social activities, individual OSN users tend to have diverse behavior patterns. We illustrate the diversity with an example. We randomly pick two users from our data set and present the most significant factors of each user’s behavioral features, side-by-side, in Table 3.1. From this table, we can observe that nearly every behavioral feature of these two users differs, implying that it is possible to tell behaviors of two users apart by comparing those feature values.

3.4 Profiling Social Behaviors

In this section, we first detail the formation of a user social behavioral profile using our proposed behavioral features. Based on our Facebook measurement study, we quantify Facebook user behavior patterns into a set of eight fine-grained metrics that correspond to the eight social behavioral features. The social behavior profile of an individual user can thus be built by combining the respective social behavioral metrics. Then, we describe the application of social behavior profiles in differentiating users and detecting compromised accounts.

3.4.1 Facebook User Behavioral Profile

In order to quantify user social behavior patterns for a specific OSN, we must first convert the social behavioral features into concrete metrics. We apply our knowledge gained in the Facebook measurement study, and devise a quantification scheme for each behavioral feature as the following.

- The *first activity* metric is defined as a 29-element vector, with each element cor-

responds to an extroversive activity on Facebook. The value of each element is the empirical probability a user engages in the associated activity as the first extroversive activity in a browser session.

- The *activity preference* metric is also a 29-element vector, similar to the *First Activity* metric. The value of each element is the empirical probability a user engages in the associated activity throughout a browser session.
- The *activity sequence* metric is defined as a 29×29 -element vector. If we conceptually arrange the vector as a 29-by-29 matrix, each cell of the matrix represents a transition between two Facebook extroversive activities $a_1 \rightarrow a_2$, whose indices are reflected by the row or column number of the cell. The value of each cell is the probability of a user to transit to activity a_2 after activity a_1 .
- The *action latency* metric is defined as an 11-element vector, and it records the empirical probability distribution of delays between consecutive HTTP requests while a user performs extroversive activities. The initial duration is zero, the first ten elements are one-second-wide bins, and element eleven is an infinite-width bin.
- The *browsing preference* metric is defined as a 9-element vector. Each element corresponds to a type of webpage on the Facebook website. The value of each element is the empirical probability a user visit the associated webpage throughout a browser session.
- The *visit duration* metric is defined as a 3×15 -element vector, and each group of 15 elements records the empirical probability distribution of the duration a user visits homepages, profile pages or application page, respectively². For each 15-element vector, we define the initial duration as zero, the first ten elements as 30-second-wide bins, the following four elements as 60-second-wide bins and the fifteenth

²We do not consider other 6 types of webpages because user visits on these pages only account for less than 8.8% of all browsing activities.

Table 3.2: A Behavioral Profile Sample

Feature	Metric Vector	Size
First Activity	[0.0, 0.0, 0.0, 0.05, 0.0, 0.0, 0.79, ...]	29
Activity Preferences	[0.09, 0.0, 0.0, 0.0, 0.0, 0.0, 0.46, ...]	29
Activity Sequence	[[0.01, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...], [0.0, ...], [0.0, ...], ...]	29×29
Action Latency	[0.17, 0.33, 0.18, 0.14, 0.11, 0.01, ...]	11
Browsing Preferences	[0.34, 0.59, 0.0, 0.01, 0.0, 0.04, ...]	9
Visit Duration	[[0.05, 0.03, 0.02, 0.04, 0.02, 0.03, ...], [0.06, 0.1, ...], [0.04, 0.03, ...]]	3×15
Request Latency	[[0.09, 0.06, 0.03, 0.02, 0.1, 0.04, ...], [0.01, 0.05, ...], [0.08, 0.08, ...]]	3×11
Browsing Sequence	[[0.05, 0.21, 0.0, 0.0, 0.0, 0.05, ...], [0.12, ...], [0.01, ...], ...]	9×9

element as an infinite-width bin.

- The *request latency* metric is a threefold 11-element vector, and each group of 11 elements records the empirical probability distribution of delays between consecutive HTTP requests during a user’s visits to homepages or profile pages or application pages, respectively. Similar to the *Action Speed* metric, the initial duration is zero, element one through ten are one-second-wide bins, and element eleven is an infinite-time-width bin.
- The *browsing sequence* metric is defined as a 9×9 -element vector. Similar to the *Activity Sequence* metric, we conceptually arrange the vector as a 9-by-9 matrix, and each cell of the matrix represents a transition between browsing two types of Facebook webpages $p_1 \rightarrow p_2$, whose indices are reflected by the row or column number of the cell. The value of each cell is the probability of the user to switch to type p_2 after finish browsing page type p_1 .

With concrete behavioral metrics in hand, we build a Facebook user’s social behavioral profile by first combining their social behavior metrics into an 8-vector tuple, then normalizing each vector so that the sum of all elements in a vector equals to one. In particular, the *visit duration* and *request latency* vectors are multiplied by a factor of 1/3; the *activity sequence* vectors and the *browsing sequence* vectors are multiplied by 1/29 and 1/9 respectively, while all other metrics are unchanged. Table 3.2 shows a sample of a Facebook user social behavioral profile.

3.4.2 Differentiating User Behaviors

The social behavioral profile depicts various aspects of a user’s online social behavior patterns, and it enables us to quantitatively describe the differences in distinct user social behaviors. In the following, we first describe how to compare social behavioral profiles. Then, we discuss the application of social behavioral profile comparison to distinguishing different user’s behaviors.

3.4.2.1 Comparing Behavior Profiles

Given two social behavioral profiles, P and Q , we quantify their *difference* in two steps.

In the first step, we compare each of the eight vectors in P against the respective vector in Q . Particularly, we measure the Euclidean distance to quantify the difference between the two vectors. Given two vectors $A = (a_1, a_2, \dots, a_n)$ and $B = (b_1, b_2, \dots, b_n)$, the Euclidean distance between them is calculated by

$$E(A, B) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}.$$

Comparing all eight vectors yield an eight-element Euclidean distance vector (E_1, E_2, \dots, E_8) . Each element in this vector has a range of $[0, \sqrt{2}]$, because the sum of each vector's elements is one.

In the seconds step, we take the Euclidean norm of the Euclidean distance vector,

$$D(P, Q) = \sqrt{\sum_{j=1}^8 (E_j)^2}.$$

The resulting value is the difference of the two behavioral profiles, and has a range of $[0, 4]$ —the more significant two profiles differ, the larger the value.

3.4.2.2 Applying Profile Comparison

To apply the profile comparison technique for differentiating user behaviors, we must further introduce another concept, *self variance*, in addition to the profile *difference*.

With two or more distinct pieces of behavioral data (i.e., clickstreams) collected from the same user, the social behavioral profiles built from each piece of behavioral data are not identical. The reasons for the differences are twofold. First, human behaviors are intrinsically non-deterministic, therefore a small amount of variation is expected even for the same activity performed by the same user. Second, because the social behavioral profile is built on top of statistical observations, errors always exist for a finite amount of samples.

A user's average behavior variance is presented. Given a collection of social behavioral profiles $\{P_1, P_2, \dots, P_n\}$ for a user U , we define the self variance of U as the mean differences between each pair of profiles:

$$V_U = \frac{\sum_{j=1}^n \sum_{k=1, k \neq j}^n D(P_j, P_k)}{n(n-1)}.$$

The corresponding standard deviation of those differences is denoted as $stdDev(V_U)$. Thus, with a probability of 97%, U 's behavior variance falls into $[0, V_U + 2 * stdDev(V_U)]$.

3.4.2.3 Detecting Compromised Account

Together with the self variance, we can apply profile comparison to distinguish different users and detect compromised accounts. Given a user U 's behavioral profile P_U , self variance V_U , $stdDev(V_U)$, and an unknown social behavioral profile Q , we can decide that the behavioral profile Q is not user U 's if the difference $D(P_U, Q)$ is larger than $V_U + n * stdDev(V_U)$, in which n is adjustable. A large n would result in a large false negative rate, while a small n would lead to a large false positive rate.

After building a user's behavior profile and variance during a training phase, we can decide whether the user's account is compromised. While the method illustrated before can be employed to fulfill the task, we adjust the method by personalizing the computation of difference to each user's behavior profile.

During the training phase, we first examine the authentic user U 's consistency on each behavior feature. Given a set of U 's clickstreams, the corresponding behavior profiles can be built as Section 3.4.1 depicted. Then we calculate the average Euclidean distance on each feature vector in U 's behavior profiles. The eight features are sorted according to the average distances in an ascending order; then each feature is assigned a weight that is inversely proportional to its rank. The weight on each feature is denoted as w_1, w_2, \dots , or w_8 . Then $D_U(P_U, Q) = \sqrt{\sum_{j=1}^8 w_j (E_j)^2}$ is employed to compute an unknown behavior profile's difference to $P(U)$.

Giving a weight on each feature is to portray a user's degree of consistency on different behavior features, which is also difficult to feign. User consistency on behavior features differs from one to one. The personalized weight on each feature in the training phase enlarges the distance in user differentiation. Heavy-weighted behavior

features that a user behaves more consistently on play more important roles in detecting impostors than light-weighted features. If an unknown behavior profile belongs to U , it is likely that its distance on heavy-weighted features are smaller than that on light-weighted features. For an impostor’s profile that does not hold this pattern, it is highly likely that the distance to U on heavy-weighted features is also large, which results in comparatively larger difference.

To sum up, the detection of account compromisation can be conducted as follows. During the training phase, given a collection of clickstreams from the account’s authentic user U , U ’s weights on the eight features w_1, w_2, \dots, w_8 are calculated first as previously stated; then U ’s self variance and the standard deviation of variance are calculated using the weighted difference formula D_U , denoted as V_U and $stdDev(V_u)$, respectively. U ’s behavior profile is built from the union of the clickstreams. For each incoming clickstream of the account, a behavior profile P_I is built from it; then the difference from P_I to P_U is calculated as $D_U(P_U, P_I)$. If $D_U(P_U, P_I) \geq V_U + n * stdDev(V_U)$, then it is classified as not from the authentic user, and thus, it is likely that the account is compromised. To guarantee a very low false positive rate (less than 3%), n is assigned to be 2.

As it is possible that a user’s behavior patterns change over time, the behavior profile needs to be updated periodically to accurately portray its patterns. While some online habits remain, a user’s behavior may evolve over time. To capture the change, the training phase can be repeated using a user’s latest clickstream to update a user’s behavior profile including feature weights.

In addition, when there are introduction of new services, new behavior features may need to be extracted. At the same time, multiple existing behavior features may also experience significant changes, which could be large enough to produce false alarms. This increased false alarm rate cannot be limited by weighinh potential

harmfulness. The training phase also need to be repeated in this scenario. New training data collection is required, and it may take some time for the detector to work accurately again.

3.4.2.4 Incomplete Behavior Profiles

Dependent upon user activeness in OSNs, the completeness of a user’s behavior profile varies. The incomplete behavior profiles should be specially processed while calculating the difference, considering the lack of sample activities from which metric vectors are built.

When some feature vectors are not available, they are not considered while calculating the difference; in this scenario the final difference will be normalized. For instance, if a user’s extroversive activity metric vectors are not available due to the reason that it does not conduct extroversive activities, its difference to another behavior profile only counts into the distances on the four introversive activity vectors; for normalization, the derived distance is multiplied by a factor of $4/\sqrt{2 \times 4}$ to serve as the final difference.

Furthermore, when there are rare sample activities to build a metric vector, it is taken as N/A. For example, if there are only 5 extroversive activities in a clickstream, the *activity preference* vector built from them can hardly be representative of the user’s behavior pattern. Hence, a threshold of the minimum number of sample activities should be assigned to guarantee the quality of metric vectors. Those vectors built from a lower-than-threshold number of sample activities are taken as N/A.

The varied thresholds of sample activity are assigned to different feature vectors. For *browsing preference* vector, it is possible that 15 page browsing activities are able to derive a comparatively representative vector; but for browsing sequence metric, 15 browsing transitions can hardly demonstrate illustrative transition probabilities.

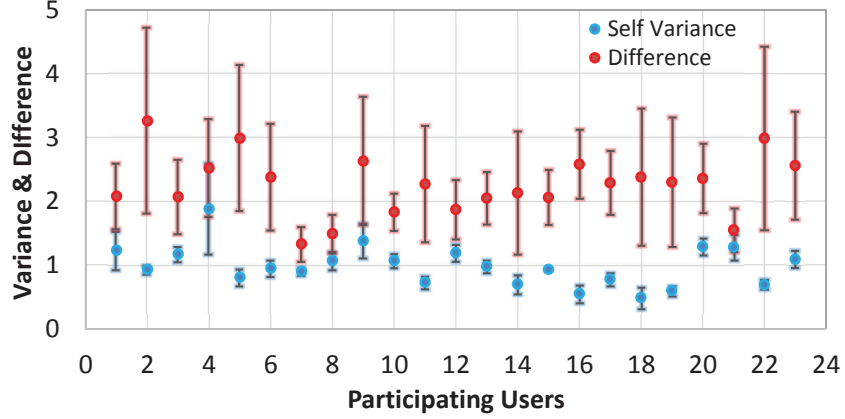


Figure 3.4: Behavior Profile Difference & Variance

Hence, when a sample threshold is assigned, it is applicable to all features except for *browsing sequence* and *activity sequence*, whose thresholds are two times of the assigned threshold.

3.5 Evaluation

We first verify that behavioral profile can accurately portray a user’s behavior pattern. Next, we validate the feasibility of employing behavioral profiles to distinguish different users, which can be used to detect compromised accounts.

3.5.1 Difference vs. Variance

We demonstrate that compared to a user’s behavior variance, its behavioral profile difference from others is more significant. That is, a user’s behavioral profile can accurately characterize its behavior pattern. We compute each sample user’s behavior variance and behavioral profile differences between its own and other users’. For each sample user, we equally partition its clickstream into four complementary parts by session, and four behavioral profiles are built accordingly; its weight on each feature

is calculated and used for difference calculation. A 4-fold cross-validation is used to calculate its average behavior variance and the average difference from the others' behavioral profiles to its profile.

Figure 3.4 shows each user's behavior variance and the average behavioral profile difference to others'. Note that only those users who have sufficient social activities, referred as "valid" users, are included in the figure. In particular, the behavioral profile of each valid user must have more than or equal to 4 non-empty feature vectors, in order to ensure that the behavioral profile is complete enough to represent the user's behavior patterns. And each feature should be derived from more than or equal to 10 social activities. For those users whose social profiles do not meet the requirement above are excluded.

As the figure shows, all users' self variance is obviously lower than its average difference from other users. This coincides with our intuition that a user's behavior variance is usually within a certain range. And comparatively complete behavioral profiles can portray users' behavior patterns. More importantly, it is possible to take advantage of the difference between behavioral profiles to discern a user.

3.5.2 Detection Accuracy

Here we further evaluate the accuracy of using social behavioral profiles to differentiate online users. We conduct three sets of experiments by varying training data size, feature quality, and profile completeness, respectively, to evaluate their impacts upon the detection accuracy.

For each sample user U , V_U and $stdDev(U)$ are calculated from its clickstream, and other users are taken as impostors. For each impostor, we calculate its behavioral profile difference to U 's using U 's weights on each feature. If the difference is larger

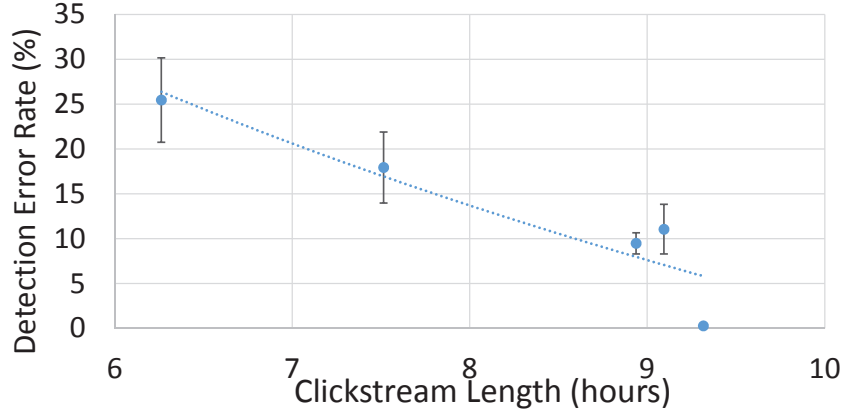
than $V_U + 2 * stdDev(U)$, the decision is taken as correct; otherwise, it is taken as a detection error. The average error rate is calculated in each scenario. Setting the threshold to be $V_U + 2 * stdDev(U)$ guarantees that U 's behavioral profiles can be discerned with the probability of more than 97%.

3.5.2.1 Input Size vs. Accuracy

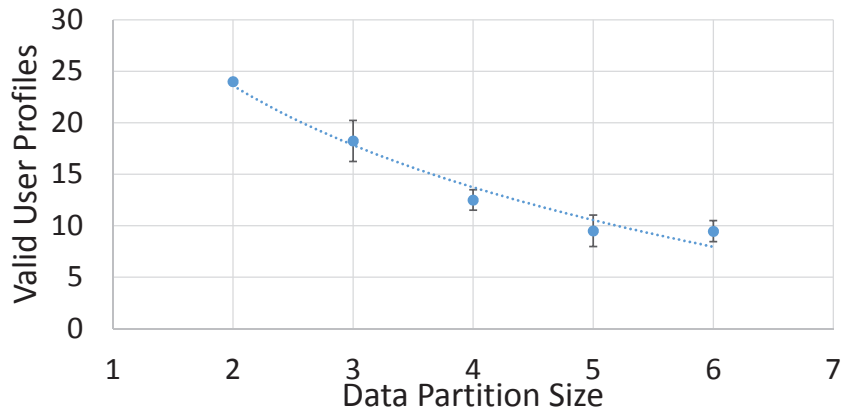
Intuitively, the more training data are given to build a user's behavioral profile, the better the profile reflects the user's behavior pattern; and hence, the profile difference demonstrates the dissimilarity between user behaviors more accurately.

We build each user's behavioral profile using the clickstream from 1/6, ..., and 1/2 of its total sessions, respectively, and use cross-validation to compute and compare behavioral profile differences. Take the 1/6 of sessions as an example, each user's clickstream is partitioned into 6 parts while the first part includes the clickstream from the 1st, 7th, 13th, ..., sessions; the second part includes the clickstream from the 2nd, 8th, 14th, ..., sessions etc. Six behavioral profiles are built accordingly and each profile is used for difference calculation. For user A , when we use the i th part of its clickstream to build its behavioral profile, the behavioral profile difference from another user B to user A is calculated six times, each of which considers A 's behavioral profile and one of B 's behavioral profiles, which is built from one out of its 6 clickstreams.

Cross-validation is used to make sure that each part of data are used for both training and validation, and the result is not derived from biased data. Furthermore, we only consider users whose behavioral profiles consist of more than or equal to 5 non-empty feature vectors, each of which should be built from more than or equal to 15 sample activities. The thresholds are set to guarantee the vector quality as well as the completeness of the behavioral profiles. To give more straightforward impression



(a) Impact of Training Data Size to Accuracy



(b) Valid Users under Varied Training Data Size

Figure 3.5: Impact of Training Data Size

over the size of clickstream we calculate its average active hours in each partition.

Figure 3.5(a) shows the dynamics of the error rate with the change of the clickstream length, while Figure 3.5(b) shows the number of valid users with different data partitions. Overall, the longer is the clickstream, the more accurate is the detection. When the clickstream is up to 10.1 hours, the error rate can reach 7.8% while longer clickstreams derive better result. When it is more than 13 hours, the error rate is as small as 0.5%. Longer clickstream provides more empirical behavior data of a user, which enables us to build more accurate and complete behavioral profiles, and hence the distance on each behavioral feature can be measured more accurately. Thus,

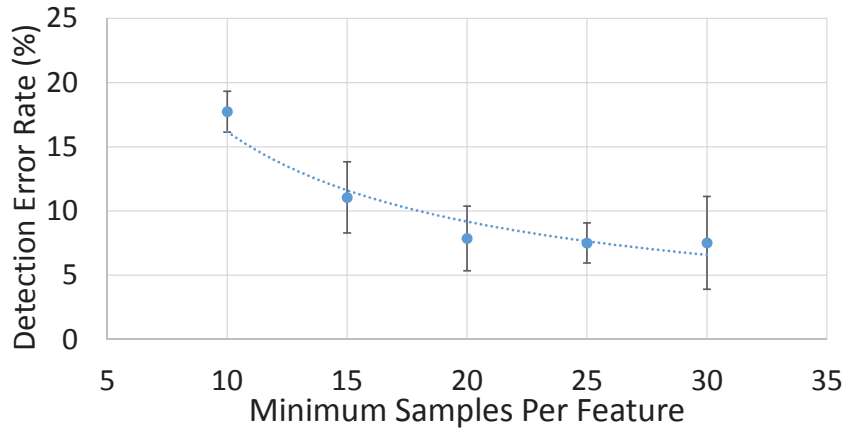
the detection is more accurate. On the other hand, with the finer partition of the clickstream, the fewer activities in each partition, leading to the smaller number of non-empty vectors and the smaller number of valid users.

3.5.2.2 Feature Quality vs. Accuracy

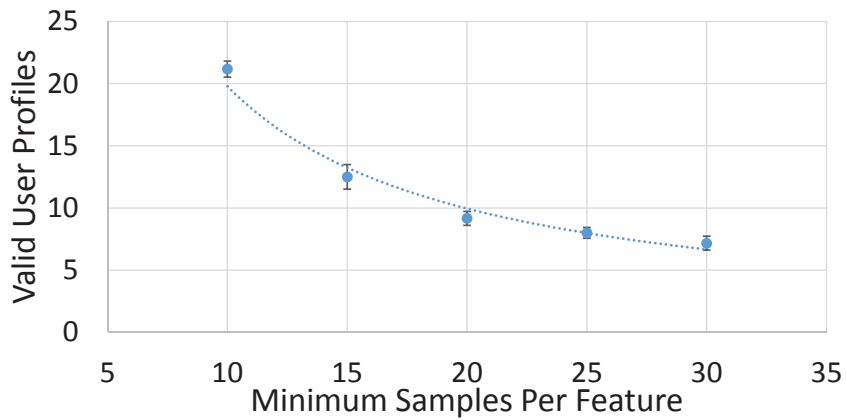
We adjust the threshold of the number of sample activities to explore whether the feature vector quality affects the detection accuracy.

When building a user's behavioral profile, the number of sample activities that derive a feature vector determines whether the feature vector represents a user's behavior accurately. By assigning a threshold to the number of sample activities, we can take control over the quality of feature vectors. We designate those vectors derived from insufficient activities to be N/A. Intuitively, higher sample activity threshold results in feature vectors with higher quality, which reduces the noise of behavior variance introduced by rare sample activities. Thus, the difference between users can be discerned more accurately.

The detection accuracy is evaluated when we assign the sample threshold to be 10, 15, 20, 25, and 30, respectively. We use the 1/4 partition of clickstreams to build user behavioral profiles and those users with less than 4 feature vectors are excluded. Same as before, 4-fold cross-validation is used to derive the average detection accuracy. The error rate with different sample threshold is depicted in Figure 3.6(a). It verifies our intuition that higher sample threshold generates feature vectors with higher quality, which helps to differentiate behavioral profiles. The side-effect to set high sample threshold is that with limited clickstream, fewer feature vectors can be built, resulting in fewer valid users, as Figure 3.6(b) shows.



(a) Impact of Feature Quality to Accuracy



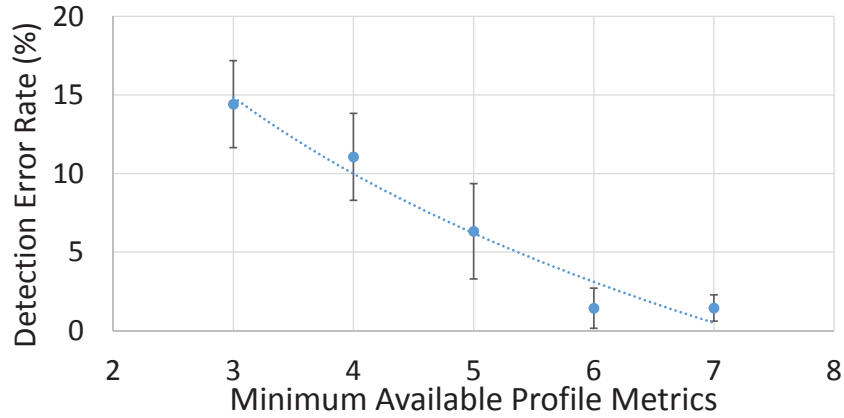
(b) Valid Users under Varied Feature Quality

Figure 3.6: Impact of Feature Quality

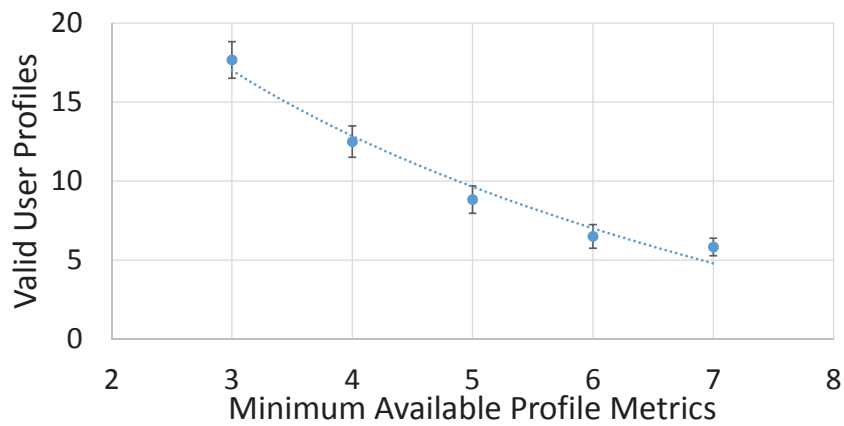
3.5.2.3 Profile Completeness vs. Accuracy

Due to the lack of certain activities, some behavioral feature vectors can be N/A. For instance, when one never conducts extroversive activities in its clickstream, at least four of its feature vectors are N/A, which makes its profile incomplete. By adjusting the least number of non-empty features vectors, the completeness of selected behavioral profiles can be guaranteed.

Here we adjust the threshold of the vector number to be 3, 4, 5, 6, 7, respectively, to examine the impact of behavioral profile completeness upon the detection accuracy.



(a) Impact of Profile Completeness to Accuracy



(b) Valid Users under Varied Profile Completeness

Figure 3.7: Impact of Profile Completeness

Same as before, the behavioral profile is built from the clickstream in 1/4 of sessions and the sample threshold is 15. A 4-fold cross-validation is conducted. The detection error rate with the change of the threshold of vector number is shown in Figure 3.7(a), while Figure 3.7(b) shows the number of valid users in each case. Evidently, more complete behavioral profiles result in higher detection accuracy, which validates the effectiveness of the behavioral features for user behavior characterization. With 7 feature vectors available, the detection accuracy reaches 100%.

Overall, active users can be distinguished more accurately by their behavioral profiles compared to inactive users. The more types of activities a user conducts,

the more complete its behavior profile can be. In addition, the more activities a user conduct in OSN, the more sample activities can be obtained within a certain duration, leading to more accurate behavioral profile.

3.6 Discussion

In this section, we first describe how social behavior profiling can be applied to detect abnormal behaviors of compromised accounts. Then we discuss adaptations of our technique to special OSN accounts (such as multi-user shared accounts and public social media accounts). We also discuss how to handle those accounts created by normal users but rarely used. Finally, we discuss the applicability of our approach to non-user-interactive channels and its limitation.

3.6.1 Detecting Compromised Behaviors

By building a behavioral profile along with the self variance for an account, we can decide whether incoming clickstreams from the account is from the authentic user or a different user. If an incoming clickstream is from the authentic user, to some extent it should comply with the behavior pattern represented by the account's behavior profile.

When an account is compromised and its behavior is well-optimized to post spam, the detection accuracy should be higher than that of differentiating another normal user. With a clear objective, to broadcast spams, spammers usually act goal-orientedly [97]. Compromised accounts can be well-programmed to focus on posting spams. Thus, their behaviors evidently deviate from common users' behaviors that are spontaneous and out of interests. As a result, the higher chance is that a clickstream consists of aggressively posting activities largely diverging from the account's

behavior profile built from unprompted activities.

Compared to an account’s evident social characteristics, such as language and interests, its social behavior is harder to feign. Even though spammers do not adopt aggressive strategies to post spams and manipulate compromised accounts to browse randomly or slow down the post speed to look normal, it is hard for them to obtain the authentic user’s unconscious social behavior pattern, not to mention to feign.

Note that our method is not limited to detect compromised accounts that are manipulated to spread spams in a specific way, such as post on the message board. Since most existing spam detection methods are tied to detect spam messages [47], [44], [90], if spam is embedded in other media, such as photos or shared files, those schemes are not applicable for detection anymore. However, using social behavior profiles to detect compromised accounts is independent from the forms of spams.

Moreover, our method can be adopted in combination with existing schemes to battle against account hijacking. In comparison with existing detection approaches, either URL or message content analysis based, our social behavior based method discerns compromised accounts from very different perspectives. Our method can serve as a complementary measure to existing detection solutions.

3.6.2 Handling Special Accounts

Some special accounts should opt out this compromisation detection scheme voluntarily in advance. Although an OSN account is normally owned by an individual user, it happens that an account is shared by multiple users. In this case, the account’s behavior variance can be much larger than that of an account managed by a single user. On one hand, such a shared account could be wrongly classified as a compromised account. (i.e., producing a false positive). On the other hand, if its behavior

variance was employed as the standard to detect account compromise, the false negative rate would be larger than using that of a single-user account.

Some public page accounts should opt out too, such as the public pages registered by business companies to promote their products. Their behaviors are also different from those of normal users since they are also goal-oriented. It is possible that those accounts are only manipulated to post ads or news. Thus, their behavior patterns are more similar to spammers than normal users.

3.6.3 Rarely Used Accounts

There exist some normal inactive OSN accounts, i.e., those accounts are created by normal users but are rarely used after the creation. Since these accounts are inactive most of time, it is hard to obtain their complete social behavioral profiles. However, on one hand, we can still build a profile for such an inactive account as long as its owner logs in at least once. On the other hand, it would be more straightforward to detect the compromise of such an inactive account because we can simply employ the existing solutions, such as checking its posting message behavior and message content, for anomaly detection.

3.6.4 Applicability and Limitation

Our method can be easily adopted to most of existing popular OSNs besides Facebook such as Google+, Twitter, and Instagram. The users of those sites are enabled to conduct various extroversive and introversive behaviors. As long as the behaviors are categorized, each user behavior profile can be built based on the eight behavior features we proposed in Section 3.3.1. Therefore, our approach can be readily applied for detecting compromised accounts in those OSN sites.

Our method is applicable to accounts whose social behavior patterns can be profiled, i.e., users who access OSN services through the Web. For normal users who directly visit the OSN webpage, their behavioral profile can be easily built via click-stream. On the other hand, it is hard to trace the behavior patterns of users who access an OSN solely via APIs, thus, our method may not be applicable for those rare cases.

However, if a compromised account uses APIs to post spams aggressively with zero social activities, we can easily detect such a compromised account given its authentic user accesses the account via the Web.

Additionally, our method assumes that cyber criminals cannot easily obtain target users' online social behavior patterns. However, if more arduous hackers compromise the physical machines that users own, they are able to learn their social behavior patterns and mimic the authentic users' social behaviors to avoid our detection. However, this requires more resourceful and determined attackers and costs much more resources and time, especially in large scale campaign.

3.7 Summary

In this chapter, we propose to build a social behavior profile for individual OSN users to characterize their behavioral patterns. Our approach takes into account both extroversive and introversive behaviors. Based on the characterized social behavioral profiles, we are able to distinguish a users from others, which can be easily employed for compromised account detection. Specifically, we introduce eight behavioral features to portray a user's social behaviors, which include both its extroversive posting and introversive browsing activities. A user's statistical distributions of those feature values comprise its behavioral profile. While users' behavior profiles diverge, individ-

ual user's activities are highly likely to conform to its behavioral profile. This fact is thus employed to detect a compromised account, since impostors' social behaviors can hardly conform to the authentic user's behavioral profile. Our evaluation on sample Facebook users indicates that we can achieve high detection accuracy when behavioral profiles are built in a complete and accurate fashion.

The Security of Twitter Trending

Twitter trends, a timely updated set of top terms in Twitter, have the ability to affect the public agenda of the community and have attracted much attention. Unfortunately, in the wrong hands, Twitter trends can also be abused to mislead people. In this chapter, we attempt to answer the question of whether Twitter trends are secure from the manipulation of malicious users. We collect more than 69 million tweets from 5 million accounts. Using the collected tweets, we first conduct a data analysis and discover evidence of Twitter trend manipulation. Then, we study at the topic level and infer the key factors that can determine whether a topic starts trending due to its popularity, coverage, transmission, potential coverage, or reputation. What we find is that except for transmission, all of factors above are closely related to trending. Finally, we further investigate the trending manipulation from the perspective of compromised and sybil accounts and discuss countermeasures.

4.1 Introduction

The Internet has subverted the autocratic way of disseminating content by traditional media like newspaper. Online trends have especially differed from traditional media

as a method for information propagation. Google Hot Trends ranks the searches that have recently experienced a sudden surge in popularity [49]. Meanwhile, these trends may attract much more attention than before due to their appearance on Google Hot Trends.

More recently, Online Social Networking (OSN) like Twitter has inaugurated a new era of “We Media.” Twitter is a real-time microblogging service. Users broadcast short messages no longer than 140 characters (called *tweets*) to their followers. Users can also join global conversations on a variety of topics at will. The topics that gain sudden popularity are surfaced by Twitter as a list of *trends* (also known as *trending topics*) [78]. Twitter and Google trends have become an important tool for journalists. Twitter in particular is used to develop stories, track breaking news, and assess how public opinion is evolving in the breaking story. Taking election campaigns as an example [83], journalists, campaigns, and pundits have tracked trends in Twitter traffic to determine candidates’ popularity and predict likely election outcomes [58].

Previous research has studied trend taxonomy [62, 76, 113], trend detection [59], and how to extract real events from Twitter trends [25, 34]. However, researchers have paid little attention to Twitter trend manipulation. It is reported that manipulators created Google trends by simply asking enough people to visit Google and search for a specific keyword phrase [10]. Also, Just *et al.* [58] inspected Twitter manipulation in an election campaign. As reported in *The Wall Street Journal*, robots have been used to undermine the “trending topics” on Twitter [23]. Thus, the focus of this work is on the issue of Twitter trend manipulation.

In this chapter, the primary questions we attempt to answer are whether the malicious users can manipulate the Twitter trends and how they might be able to do that? By answering these questions, we can also gain insights into how to enhance a commercial promotion campaign by reasonably using Twitter trends. To investigate

the likelihood of manipulating Twitter trends, we need to deeply understand how Twitter trending works. Twitter states that trends are determined by an algorithm and are always topics that are immediately popular. However, the detailed trending algorithm of Twitter is unknown to the public, and we have no way to attempt to find out what it is. Instead of working out the detailed trending algorithm, we study Twitter trending at the topic level and infer the key factors that can determine whether a topic trends from its popularity, coverage, transmission, potential coverage, and reputation. After identifying those key factors that are associated with the trends, we then investigate the manipulation and countermeasures from the perspective of these key factors.

The major contributions of this work are as follows:

- We demonstrate the evidence of the existing manipulation of Twitter trends. In particular, employing an influence model, we analyze the dynamics of an endogenous hashtag and identify the manipulation from the endogenous spread. Then, centering on a spike in the dynamics of a topic, we compare topic's influence before and after the spike and investigate the accounts in the spike. We can see the existence of a suspect spamming infrastructure.
- We study Twitter trending at topic level, considering topics' popularity, coverage, transmission, potential coverage, and reputation. The corresponding dynamics for each factor above are extracted, and then Support Vector Machine (SVM) classifier is used to check how accurately a factor could predict the trending. We find that, except for transmission, each studied factor is associated with trending. We further illustrate the interaction pattern between malicious accounts and authenticated accounts, with respect to the trending.
- We present the threat of malicious manipulation of Twitter trending, given

compromised and sybil accounts in the suspect spamming infrastructure we observed. Then we demonstrate how compromised and sybil accounts could threaten Twitter trending by simulating the manipulation of dynamics as compromised and sybil accounts would do. Countermeasures are then discussed to defend against the manipulation of Twitter trending.

In this chapter we validate our dataset firstly. Then we demonstrate the evidence of manipulation of Twitter trends and the key factors that determine Twitter trending are explored. At last we discuss the threat of manipulation of Twitter trends and corresponding countermeasures.

4.2 Dataset

4.2.1 Data Collection

We collected our dataset via *Twitter API* through two different collection windows. One lasted 40 days, from June 6, 2013, to July 15, 2013, and the other lasted 30 days, from August 26, 2013, to September 26, 2013. At the end, we obtained more than 69 million tweets from 5 million accounts. Since we focus on the hashtags, we only analyze the tweets with hashtags. More specifically, our dataset was collected via Stream API. We also collected the public trends of Twitter via Rest API.

Sample Stream and Search Stream. We obtain a sample stream via Twitter’s Streaming API. We define the 15 most frequent hashtags in the sample stream as *sample trends*. Sample trends are retrieved from the sample stream every 30 minutes. We create a search stream by opening up a new streaming channel via Streaming API and searching sample trends. Therefore, the sample stream and search stream are not inclusive of each other, since they are from two different streaming channels

of the Streaming API.

Public Trends and Sample Trends. Twitter trends include trending hashtags and trending keywords. Our focus is on the trending hashtags. Thus, the trends in the rest of the chapter represent trending hashtags only. Public trends are published by Twitter and available via the Twitter API. Sample trends are obtained by ranking the frequency of hashtags over the sample stream. Note that, throughout this chapter, trends represent public trends if not specified. The trends used to conduct trending analysis are the intersection of sample trends and public trends.

Sample Dynamics and Search Dynamics. We define the *dynamics* of a topic as the variation of the topic against time with respect to a specific frequency feature, such as tweet number or account number. For a certain topic, we obtain its dynamics through its sample stream and search stream independently. Sample dynamics represent how the topic evolves in the sample stream, while search dynamics reflect the evolution of the topic in the search stream.

4.2.2 Validation of Dataset

The major objective of this work is to study the key factors of Twitter trending and inspect the possible manipulation of these factors. In this respect, we validate the representativeness of our dataset in two ways. On one hand, sample trends are supposed to reflect the public trends to a certain extent; on the other hand, the syncretization of sample dynamics and search dynamics should be able to embody the critical information for inferring the key factors of Twitter trending.

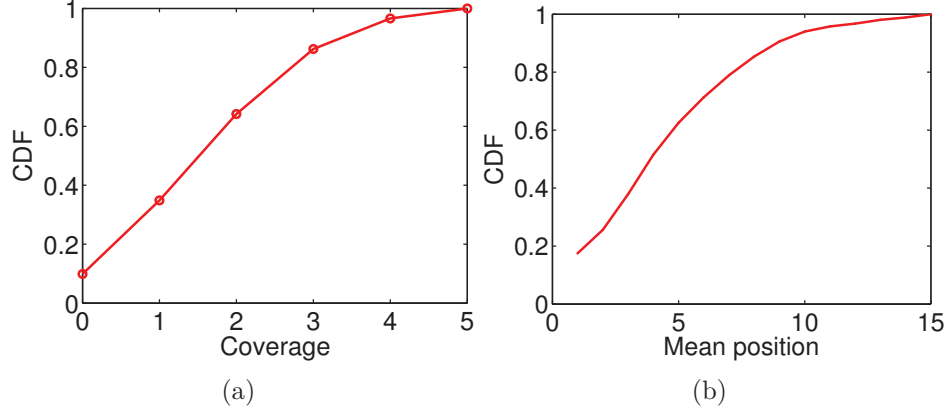


Figure 4.1: Coverage and Mean Position of Sample Trends

4.2.2.1 Could sample trends reflect public trends?

Sample trends are the 15 most frequent hashtags of a sample stream. They are used as query keywords to profile topic dynamics. Topic dynamics are then used to infer the key factors of Twitter trending. If sample trends could not reflect public trends, the collected topics' dynamics would be meaningless for studying the key factors of public trends.

We employ *coverage* and *mean position* to test whether sample trends reflect public trends. Coverage is defined as the number of hashtags that are common in both sample trends and public trends, and mean position represents the average rank of the common hashtags in sample trends. Therefore, coverage can be expressed as

$$Coverage = \{Sample\ trends\} \wedge \{Public\ trends\}. \quad (4.1)$$

Recall that we collect 15 sample trends and there are 5 hashtags in the public trends. Therefore, coverage is equivalent to or less than 5, and mean position is between 1 and 15. Fig. 4.1(a) and Fig. 4.1(b) show the coverage and mean position of the sample trends respectively. We observe that more than 90% of the sample trends

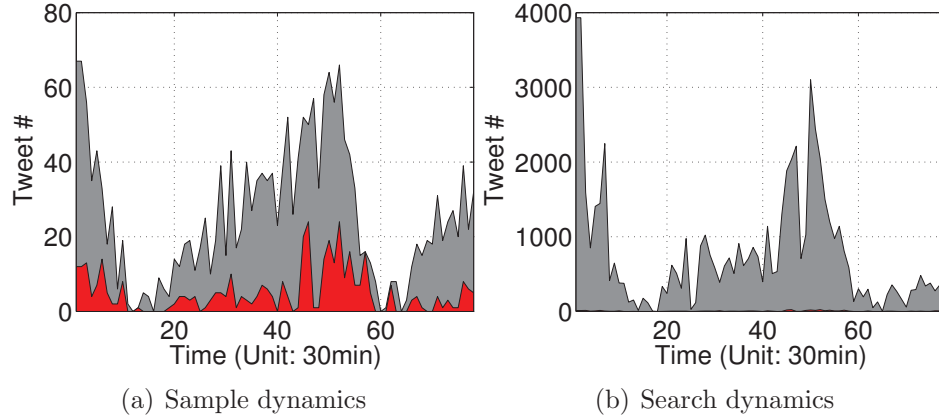


Figure 4.2: Sample and Search Dynamics

have at least one common hashtag with public trends and almost 60% of them rank the common hashtags as the top 5 trends. It suggests that the sample trends of our dataset reflect the public trends.

4.2.2.2 Could observed dynamics reflect general dynamics?

Whether the sample dynamics and the search dynamics we collect could reflect the general dynamics is critical to determine whether our observed dynamics could be used to infer the key factors of Twitter trending. Here we define the general dynamics of a topic as the dynamics that contain the whole collection of tweets related to the topic. However, the general dynamics of Twitter are well beyond the reach of most researchers. Thus we compare the sample dynamics and search dynamics instead using the Jensen-Shannon divergence metric [66].

We collect data from both streaming API and search API and obtain sample and search dynamics respectively. Both sample dynamics and search dynamics are samples of general dynamics. Morstatter *et al.* [75] demonstrated that sample data from streaming API could represent the overall data to some extent. We compute the distance in the probability distribution of sample and search dynamics using the

Table 4.1: The Jensen-Shannon Divergence

	Sample	Search	Intersection
Sample	-	0.05	0.08
Search	0.05	-	0.07
Intersection	0.08	0.07	-

Jensen-Shannon divergence metric [66]. The JensenShannon divergence metric is used to measure the similarity between two probability distributions. We randomly choose a trending hashtag “oomf”. Fig. 4.2 shows sample and search dynamics, as well as the intersection of two dynamics(red histogram shows the intersection). We compute the Jensen-Shannon divergence for sample dynamics (S_a) and search dynamics (S_e) as follows:

$$JSD(S_a \parallel S_e) = \frac{1}{2}[KL(S_a \parallel M) + KL(S_e \parallel M)], \quad (4.2)$$

where $M = \frac{1}{2}(S_a + S_e)$ and KL is the Kullback-Liebler divergence [42]. We also calculate the Jensen-Shannon divergence for sample dynamics and intersection dynamics, as well as search dynamics and intersection dynamics. Table 4.1 shows the results. We can see that none of them exceeds 0.1, especially only 0.05 for sample dynamics and search dynamics. We can infer that there is insignificant divergence between sample dynamics and search dynamics. Also, the fact that either sample or search dynamics have no significant divergence with intersection dynamics can further support endogenous relationship between sample and search dynamics. In other words, the observed dynamics are very likely to be consistent with general dynamics.

4.3 Evidence of Manipulating Topic Dynamics

In this section, we present the evidence of Twitter trend manipulation through modeling analysis and estimation of topics’ influence. Existing literature has identified

two important factors for topics becoming trends: the endogeneity that captures the propagation effect of the topic in the network and the exogeneity that represents the driving force external to the network (e.g., the mass media) [26, 76].

First, we need to distinguish manipulation from exogenous factors. In general, exogenous factors represent external and legitimate factors, especially the mass media. However, manipulation is intended either as malice or as a means to an end. But it is still impossible to quantify the difference between them. To avoid the impact of exogenous factors, we choose the hashtags that only spread inside social networks, like Twitter. Then, we employ an influence model to capture the spread due to the effect of social networks and trace out the evidence of manipulation.

4.3.1 Selecting Hashtags in Twitter

A number of hashtags always flourish in Twitter. Some of them do not correspond to external events (e.g., an earthquake). We call these endogenous hashtags *memes* throughout this chapter. Most of the memes are combinations of words or acronyms, which are used to express an emotion or raise a question. Since the memes are not associated with any external events, the spread of the memes can be only due to the effect of social networks and manipulation. The effect of social networks could be captured by the influence model [106], while the manipulation of a meme can be regarded as the effort to drive the meme to trend beyond the effect of the network. To determine whether a hashtag is a meme, we manually check if the hashtag has been covered by any news media.

4.3.2 Endogenous Factors and Manipulation

We employ an influence model (Linear Influence Model, LIM [106]) to capture the network effect on the spread of the memes. LIM is used to model the global influence of a node (an account) on the rate of diffusion through a network, which can be expressed as

$$V(t+1) = \sum_{u \in A(t)} I_u(t-t_u), \quad (4.3)$$

where $V(t+1)$ represents the number of nodes that are influenced at time $t+1$, $A(t)$ denotes the set of nodes that have already been influenced before time t , and $I_u(l)$ is the influence function of node u at l th time step after it is influenced at time t_u ($t_u < t$). LIM has been evaluated that, for the memes mentioned above, most of the observed dynamics could be attributed to the influence of nodes, especially considering the imitation factor $b(t)$:

$$V(t+1) = \sum_{u \in A(t)} I_u(t-t_u) + b(t). \quad (4.4)$$

The imitation means that nodes imitate one another because the topic is popular and everyone talks about it. However, for the memes, the imitation happens only due to the spread in the network. Therefore, we exclude imitation from the model and take the manipulation $ex(t)$ into account. The influence model we consider is

$$V(t+1) = \sum_{u \in A(t)} I_u(t-t_u) + ex(t). \quad (4.5)$$

Extensive research has been done on the influence in Twitter [32, 39, 84, 99]. Researchers not only inspected the effectiveness of different influence measures, such as follower number, retweet number, and mention number, but also proposed algorithms

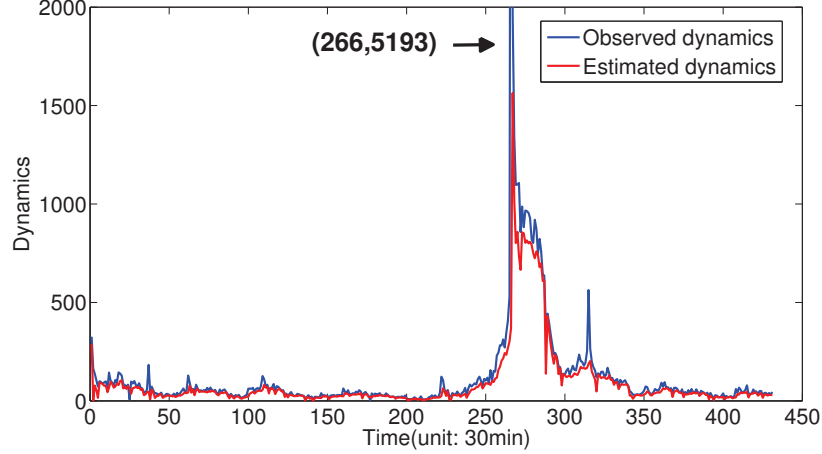


Figure 4.3: Observed and Estimated Dynamics of “ThrowbackThursday”

to measure the influence.

In this section, our goal is to demonstrate the impact of manipulation on the observed dynamics. Different from LIM, we do not consider the influence from the pointview of a single account but from the perspective of the observed dynamics. Therefore, we take the accounts that appear in the dynamics within one time slot as a single node. Each time slot is 30 minutes. The accounts that appear in the observed dynamics before time slot t , exert the influence on the accounts that appear in the dynamics within time slot t . Consequently, we can get $\sum_{u \in A(t)} I_u(t - t_u) \approx \sum_{s < t} I(V(s))$, where $I(V(s))$ denotes the influence of the accounts that appear in the dynamics at time slot s on the accounts that appear in the dynamics at time slot t . The influence of any single time slot would fade away as time passes. The influence function could be further simplified as $\sum_{K \leq i < 0} I(V(t - i))$ when only considering K time slots before time slot t . By assuming that the influence is linear to the time lag, we can further express $\sum_{K \leq i < 0} I(V(t - i))$ as a linear model, $\sum_{K \leq i < 0} V(t - i) \cdot l_i$. The parameter l_i can be estimated by least squares.

Fig. 4.3 shows the observed dynamics and the estimated dynamics from the influence model expressed in Eq. 4.5 for the meme “ThrowbackThursday.” Here, the

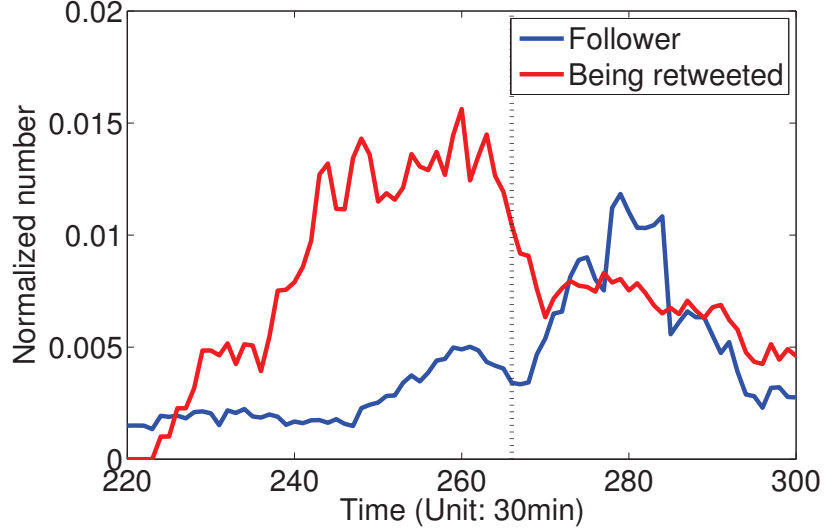


Figure 4.4: Normalized Number of Follower and Retweet for “ThrowbackThursday”

dynamics is the evolution of tweet number. For the linear model we consider, coefficient of determination R^2 can indicate the proportion of variability in the observed dynamics that may be attributed to the linear combination of explanatory variables. R^2 is calculated as 0.705 for the whole dynamics, but when we exclude the *spike* as indicated in the figure, R^2 is 0.995. It suggests that the influence in the network should be capable of explaining most of the observed dynamics except some specific spikes. In other words, there must exist other driving factors except the influence to produce the spikes. For the memes we select, the driving factors except the influence are far more likely to be manipulation than any other exogenous factors, such as news and mass media.

We further estimate the influence of each time slot upon the dynamics of a topic. The follower number of the accounts in a time slot represents the number of potential accounts that will be exposed to the topic in the following time slots, which could predict the influence of the time slot. We consider the follower number of the accounts as *pre-estimation* of influence. The number of being retweeted for the tweets in a time

slot measures to what extent the tweets in this time slot are adopted by the accounts that are exposed to the topic in the following time slots, which hence can be used to estimate the practical influence of the time slot. We regard the number of being retweeted as *post-estimation* of influence. Fig.4.4 shows the normalized number of followers and the normalized number of being retweeted for “ThrowbackThursday” around the spike. We view the number of followers and the number of being retweeted as prediction and estimation of influence, respectively. It is evident that (1) there exists a large gap between the pre-estimation and post-estimation of influence before the spike, and (2) after the spike, the post-estimation of influence falls and gets close to the pre-estimation of influence. The most likely explanation is that the manipulation before the spike leads to exceptional retweets and after the spike, the manipulation ends.

4.3.3 Investigating the Accounts in the Spike

We can verify our conjecture by investigating the accounts in the highest spike as shown in Fig.4.3. We collect their friends (i.e., the accounts that they follow) and check whether their friends have shown up in the dynamics before, or in other words, whether the accounts in the spike join the topic after their friends. For the 4,055 accounts in the spike, 63.8% of them join the topic after their friends. There are still over 1,000 accounts that do not join the topic after their friends. We could not simply make any conclusion based on the ratio of the accounts that join after their friends because the dynamics is sampled.

Nevertheless, we can further check the accounts that have been suspended by Twitter. It is intuitive to link manipulation to malicious accounts. By the time of checking accounts (about 2 months after crawling sample and search stream), 118

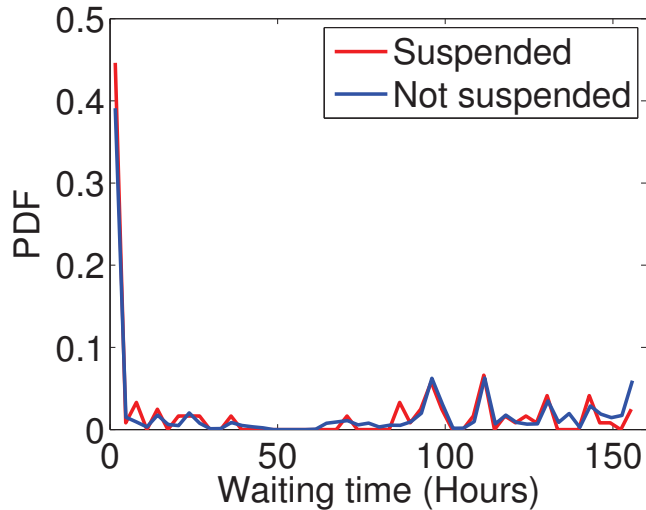


Figure 4.5: Waiting Time of Accounts in the Spike

accounts have been suspended by Twitter. We compare the temporal feature (waiting time) of suspended accounts with that of the accounts not being suspended. Waiting time means the interval from the time when an account’s friend joins the topic to the time when the account itself joins. Fig.4.5 depicts the PDF of the waiting time of suspended accounts and that of still-active accounts. It is evident that the waiting times of both kinds of accounts are mostly within one day, which is similar to the waiting times of other human activities following power-law distribution. However, the waiting times of those two kinds of accounts have the same spikes around 100 hours, implying there exist other malicious accounts that have not yet been detected by Twitter.

We further check the predecessors of the accounts in the spike, and identify the accounts that have already been suspended by Twitter. We define *descendants* of account A as those accounts that follow account A and publish at least one tweet of a certain topic. We then study the descendant number of the malicious accounts and the descendant number of their first generation and second generation, and so forth. Level 0 denotes the malicious accounts themselves. Level 1 is the first generation of

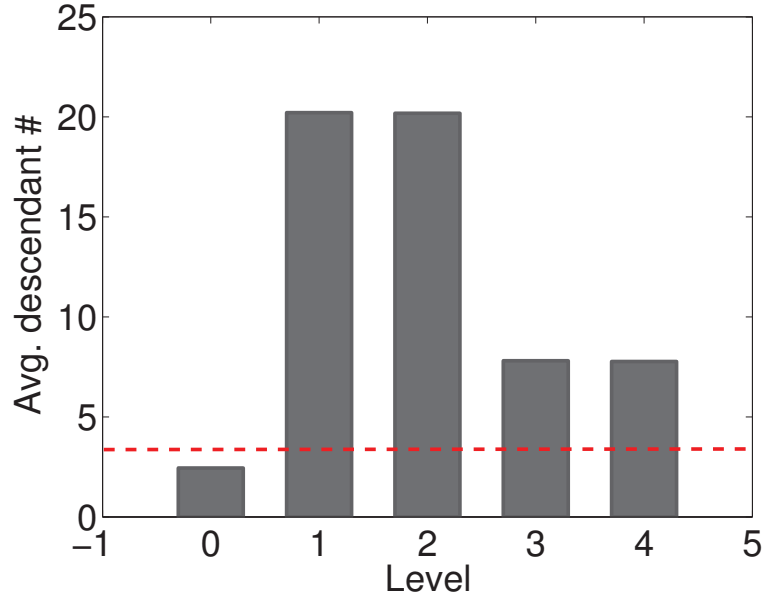


Figure 4.6: Average Descendant Number of Malicious Accounts

the malicious accounts. The rest can be deduced by analogy. Fig.4.6 demonstrates the average descendant number of five different levels starting from level 0. It is interesting that the average descendant number of the malicious accounts (level 0) is almost the same as the average descendant number of all accounts (as the red dashed line indicated). The first and second generations exhibit extraordinarily large average descendant numbers. And the descendant number falls sharply when it comes to levels 3 and 4. Since the first-generation descendants of the malicious accounts are the followers of the malicious accounts, they tend to be malicious or compromised. Specifically, malicious accounts use them to construct the spamming infrastructure. This explains why their descendant number increases sharply.

Overall, we have three observations for the manipulation involved with the accounts in the spike: (1) Variation of the topic’s influence suggests the manipulation around the spike; (2) The waiting time distribution indicates the existence of still-active malicious accounts; and (3) The descendant number indicates the existence of the suspect spamming infrastructure.

4.4 Inferring the Key Factors of Twitter Trending

After showing the suspected manipulation of Twitter trends, we proceed to infer the key factors of Twitter trending. In this section, we first syncretize sample dynamics and search dynamics to produce the syncretized dynamics. With the syncretized dynamics, we then infer the key factors that matter to trending using the SVM classification method.

4.4.1 Syncretizing Sample and Search Dynamics

Since sample dynamics and search dynamics are obtained from independent streams, syncretizing sample dynamics and search dynamics could integrate the information from both. Sample dynamics is continuous but is a smaller portion of general dynamics, while search dynamics is discontinuous and consists of a larger portion of general dynamics.

We employ a Kalman filter to generate the synthesized dynamics. The Kalman filter provides a recursive means to produce the estimation of unknown variables using a series of measurements observed over time, containing noise and other inaccuracies [14]. Since both dynamics are sampled from general dynamics, we can estimate incontinuous search dynamics from continuous sample dynamics and then treat the estimated search dynamics as the input measurements of the Kalman filter. After that, we generate a syncretized dynamics by integrating sample dynamics into search dynamics. Fig.4.7 demonstrates an example of the Kalman filter for hashtag “oomf.” We plot sample dynamics, estimated search dynamics, and the syncretized dynamics after Kalman filtering. The syncretized dynamics retain the basic features of sample and search dynamics but remove some of the noise of estimated search dynamics.

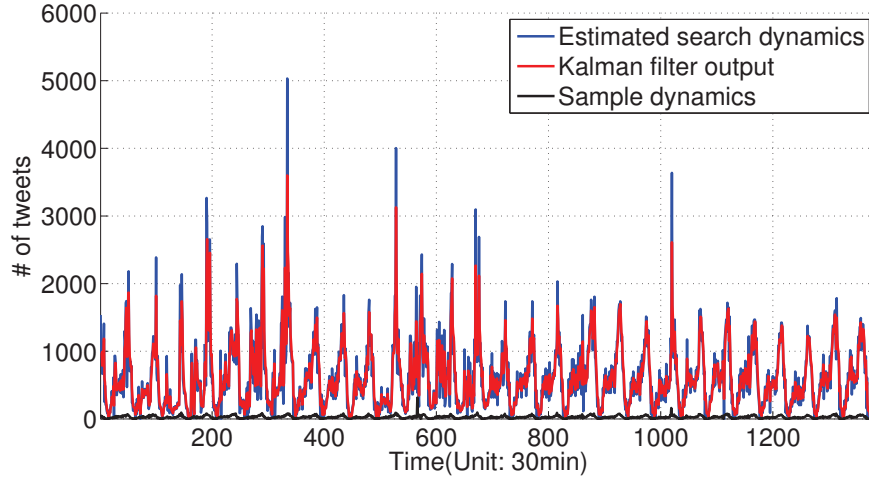


Figure 4.7: Example of Kalman Filter

4.4.2 Analyzing Key Factors of Twitter Trending

The trending algorithm processes a stream of tweets and produces trends for users. From the user’s perspective, the trending algorithm is supposed to dig out the most popular and attractive topics from the stream. To meet this demand, the trending algorithm may need to take into account some other factors besides topics’ popularity. In this section, we explore the relevance of several factors with the trending. As each factor is associated with a specific dynamics, we investigate how accurately the dynamics of a factor could predict the trending.

4.4.2.1 Segment of Dynamics

Due to the data collection method, the dynamics we obtain are naturally slotted. For a specific time point t , we assume that M time slots right before t is long enough to determine whether a topic will trend, and we define this time period as one segment. Therefore, for each time point of the dynamics, the segment right before it consists of M time slots, as Fig.4.8 shows.

Each segment corresponds to a binary sign, which indicates whether the topic

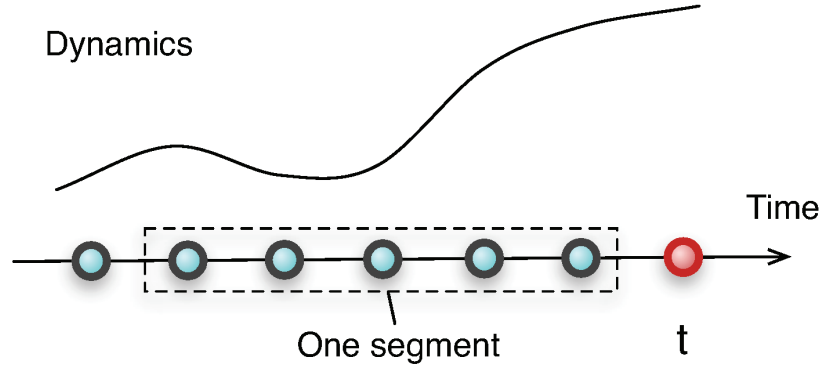


Figure 4.8: Example of One Segment

trends or not at the end of the segment. Let $\{S_i, T_i\}$ denote the i th pair of segment and binary sign, where S_i and T_i represent the i th segment and its binary sign, respectively. Next, we input a series of segments and binary signs for the SVM classifier.

4.4.2.2 SVM Classifier

We choose Support Vector Machines (SVMs) as our classifier to determine how accurately a factor could perform the binary classification. SVMs have been widely used to address many different classification problems, including handwritten digit recognition [19], object recognition [79], text classification [57], and image retrieval [95].

The basic purpose of SVMs in a binary classification problem, is to map the feature vectors into a high-dimensional space and find the optimal hyperplane that represents the largest separation, or margin, between two classes. We obtain d -dimensional feature vectors by calculating the statistics of the segments (e.g., mean, standard deviation) and get corresponding class labels from the binary signs mentioned above.

Our procedure of resolving the classification problem can be summarized as (1) conducting scale on the data, (2) choosing the RBF kernel, and (3) using cross-

Table 4.2: Topics Extracted from Datasets

Topics
ImSingleBecause
SingleBecause
tgif
20factsaboutme
wecantdateif
ifwedata
IHatePeopleThat
MentionSomeoneHandsome
mentionsomeonebeautiful
TalkAboutYourCrush
easilyattractedto

validation to find the best parameters C (penalty parameter) and γ (tunable parameter of RBF kernel) for the minimization problem and achieve the best classification accuracy. In our proof-of-concept implementation, we employ the open-source SVM package LIBSVM 3.17 [41].

4.4.2.3 Experiment Results

To examine the factors of the trending, we first extract a collection of topics from our dataset. Table 4.2 lists the topics. The topics are all *memes*, as mentioned in Section 4.3. In addition, there are similar topics in the list, such as “ImSingleBecause” and “SingleBecause.” However, we keep the similar topics apart because they all trend at least once. Note that we only extract 11 topics for the SVM classification, since the input unit for the SVM classifier is the segment in the dynamics of the topics. Each topic has more than 1,000 segments. Therefore, we can obtain more than 10,000 samples in the training set for the SVM classifier.

For each topic, we trace the dynamics of each factor we will inspect later. All dynamics are traced in the granularity of 30 minutes. The granularity of dynamics

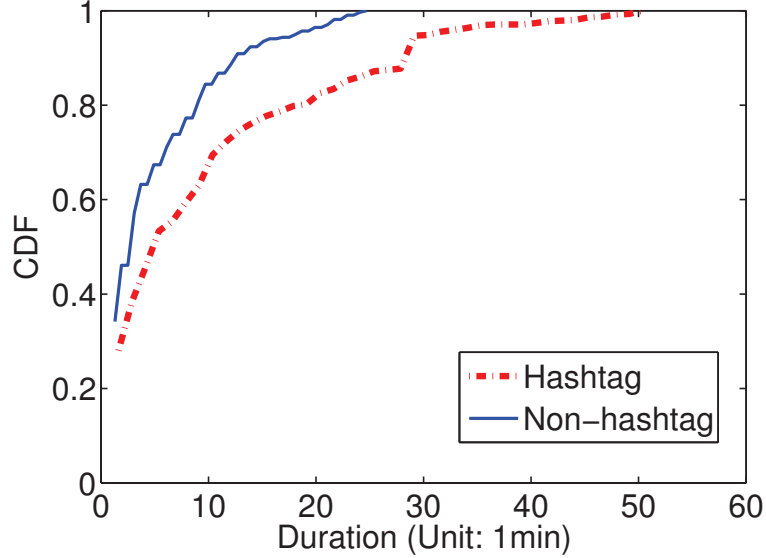


Figure 4.9: Trending Duration of Topics

is supposed to be larger than the trending duration of most topics, such that we can regard each trending as a point in the dynamics. Fig.4.9 depicts the trending duration of all trends in our dataset, including hashtag trends and non-hashtag trends. The granularity of 30 minutes we choose is larger than the duration of most trends, including both hashtag trends and non-hashtag trends. Also, we observe that hashtag trends last longer than non-hashtag trends.

After tracing the dynamics, they are divided into segments of length M . By calculating the statistics (say, mean and standard deviation) and frequency, we map each segment into a d -dimensional feature vector ($d = 16$). The corresponding indicator label is obtained from the public trends data we collect, such that we have samples composed of feature vectors and indicator labels. These samples compose the training set. More specifically, the training set is made up of positive samples (with indicator label being 1) and an equal number of negative samples (with indicator label being -1).

To quantify the extent to which a factor is associated with trending, we measure

the best classification accuracy that the factor can achieve by using a grid parameter search, a tool in LIBSVM. The best classification accuracy by employing a grid parameter search can reflect the maximal probability in which a factor is associated with the trending. Specifically, we consider the factors of a topic that impact the trending from its popularity, coverage, transmission, potential coverage, and reputation. These five factors are then operationalized with five behavioral and structural variables (tweet number, account number, mention number, follower number, and tweet history number, respectively). Corresponding dynamics are assigned to each factor. We describe the factors and the corresponding dynamics as follows:

Popularity and Tweet Dynamics. The popularity of a topic represents the topic’s vitality. We use the tweet number of a topic to capture the topic’s popularity. The tweet dynamics of a topic record the variations of the number of tweets about the topic. It is the most frequently used metric for measuring the evolution of events and detecting trending topics. The number of tweets at a specific time makes the popularity of a topic easily and directly perceived through the senses.

Coverage and Account Dynamics. Coverage of a topic means the participation rate of the topic. We can employ the account number of a topic to quantify its coverage. Account dynamics reflect the variations of the number of accounts involved in the topic. Compared with tweet dynamics, account dynamics exclude the impact of those extremely active accounts in the trending. The number of accounts at a specific time may serve as a more reliable popularity gauge for a topic than the number of tweets.

Transmission and Mention Dynamics. Transmission of a topic is the extent to which users may retweet or reply to the topic. The mention number of a topic is used to express the topic’s transmission. The mention dynamics of a topic record the variations of the number of mentions appearing in the tweets about the topic. The

mention we study includes both direct mention and retweet, since both of them use “@username.” Either direct mention or retweet can represent the means to propagate the topic. The propagation of a topic is very important for making the topic trend.

Potential Coverage and Follower Dynamics. The potential coverage of a topic represents the potential participants due to propagation of the topic on the basis of current participants. We use the follower number of a topic to capture the potential coverage of the topic. The follower dynamics of a topic are the variations of aggregate follower numbers of the accounts involved in the topic. The follower number represents the number of those accounts that the topic reaches and may join in the topic next. In general, followers play a more important role in the propagation of a topic than mentions.

Reputation and Tweet History Dynamics. The reputation of a topic is a kind of credibility that reflects whether the topic conforms to the main awareness of Twitter. We select the tweet history number of a topic to quantify its reputation. For an account, its tweet history number means the aggregate number of tweets that the account posts from its creation. The tweet history dynamics of a topic record the variations of aggregate tweet number of the accounts involved in the topic. The tweet history number of an account can reflect its reputation, which is earned by remaining active for a long time. The more historical tweets an account posted, the more audience it potentially has. Therefore, the account may be more likely to enable the trending of a topic it joins, either as a source or as a propagator of the topic.

After specifying the dynamics of each factor, we train the SVM classifier using the training set. Recall that the d -dimensional feature vector of each sample is obtained by calculating the statistics and frequency of the segments for the dynamics. Initially, we select 36 statistical and frequency features (i.e., $d = 36$). The feature set can capture all of the statistical characteristics of the dynamics. We then employ the

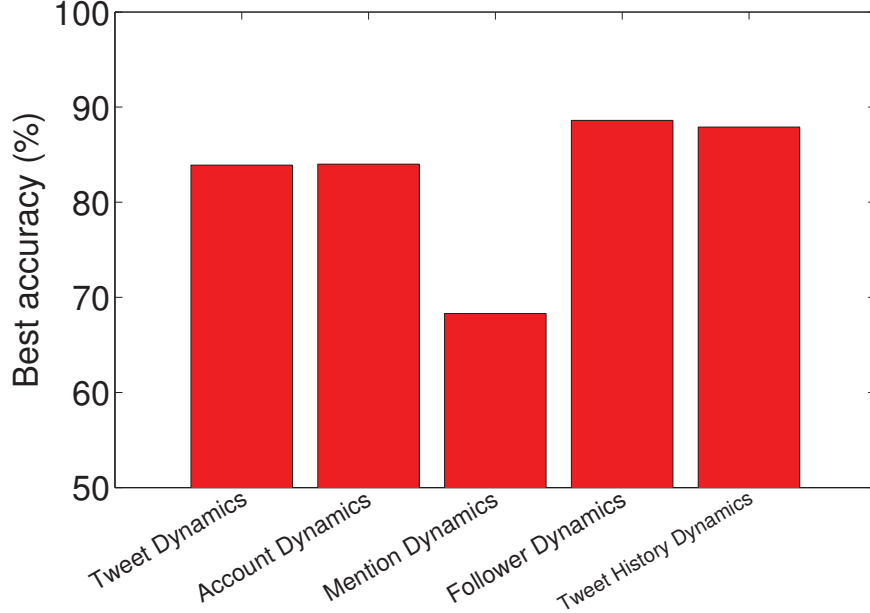


Figure 4.10: The Best Accuracy for Dynamics of Each Factor with $M = 12$

feature selection tool to extract the appropriate features for improving the classification. Specifically, we get a feature set with 16 features. Fig.4.10 depicts the best classification accuracy of each single factor. We observe that follower dynamics and tweet history dynamics are most associated with trending. Tweet dynamics and account dynamics come after but are still closely related to trending. However, mention dynamics can hardly predict trending with the best accuracy being as low as 68%.

Segment Size. We then investigate whether the best accuracy is sensitive to segment size M . We calculate the best accuracy of each factor for $M \in \{4, 8, 12, 16\}$. Fig.4.11 shows the variation of segment size ($M \in \{4, 8, 12, 16\}$). It is observed that the best accuracy slightly increases when the segment size increases for each factor. Nevertheless, the best accuracy approaches the maximum when the segment size is large enough, especially for the factors that are more closely related to trending.

Suspended Accounts vs. Authenticated Accounts. Suspended accounts and authenticated accounts exist in the account dynamics. We identify whether an

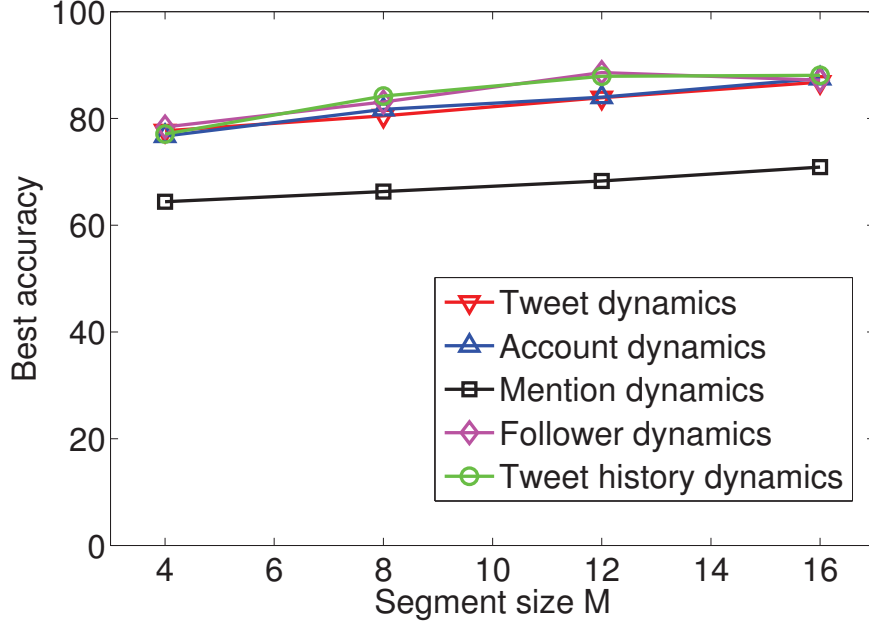


Figure 4.11: Variation of Segment Size ($M \in \{4, 8, 12, 16\}$)

account is authenticated or suspended by crawling the account’s information from its webpage on Twitter. The webpage crawling is performed about six months after collecting the dynamics, which should be enough time for malicious accounts to be detected. It is reasonable to assume that suspended accounts are malicious accounts. Malicious account dynamics could indicate the extent to which the trending is associated with malicious activity, while authenticated account dynamics reflect how closely the trending is related to the mainstream¹ of Twitter. The dynamics of a topic may be affected by the mainstream, but in the meantime, they are interwoven with the malicious activity. It is interesting to examine which of them (the mainstream and the malicious activity) is closer to the trending of the topic. Before doing that, we first explore the relationship between malicious accounts and authenticated accounts for each topic. Fig.4.12 shows the Pearson correlation coefficient of malicious accounts and authenticated accounts for the 11 topics in Table 4.2. The Pearson correlation

¹By mentioning the *mainstream*, we mean the public awareness that comes into being on Twitter due to the higher reputation of authenticated accounts.

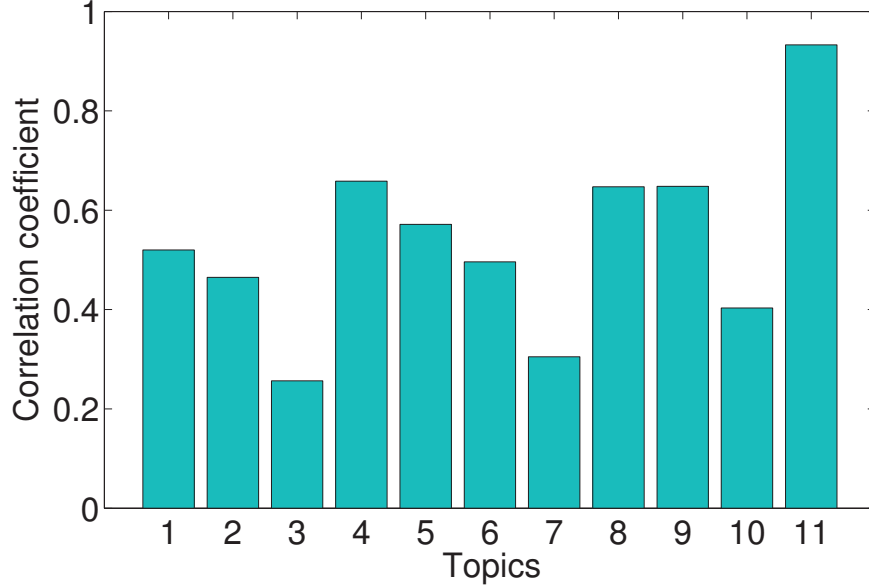


Figure 4.12: Correlation of Suspended and Authenticated Account Dynamics

coefficient (ρ) of malicious accounts (S) and authenticated accounts (A) is calculated as

$$\rho = \text{corr}(S, A) = \frac{\text{cov}(S, A)}{\sigma_S \sigma_A},$$

where *cov* means covariance, and σ is the standard deviation. We observe that all topics we studied have a positive linear relationship between malicious accounts and authenticated accounts. It may indicate the interweaving function of malicious accounts and authenticated accounts in the trending. Therefore, it is necessary to figure out which factor outweighs the others in terms of the trending.

We show the comparison of malicious and authenticated account dynamics in terms of predicting the trending in Fig.4.13. It is observed that malicious account dynamics are more closely associated with the trending than authenticated account dynamics for five topics (“tgif,” “wecandateif,” “ifwedata,” “MentionSomeoneHandsome,” and “mentionsomeonebeatiful”).

We further examine how malicious account dynamics become closely related to

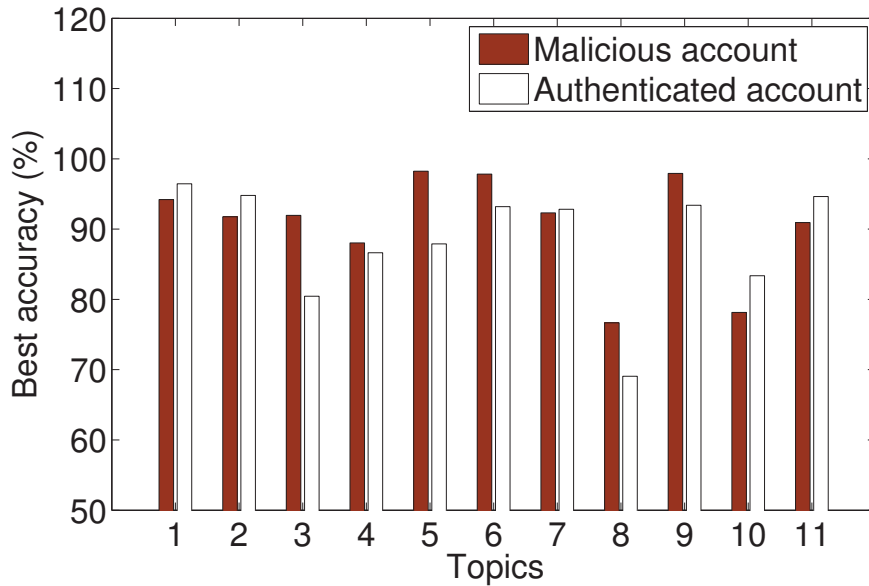


Figure 4.13: Best Accuracy of Suspended and Authenticated Account Dynamics

the trending, especially how malicious accounts interact with authenticated accounts. We extract the peaks of malicious accounts and authenticated accounts across the collection window for the five topics mentioned above. Each peak represents an intense involvement of malicious accounts or authenticated accounts. Fig.4.14 shows the malicious account peaks and authenticated account peaks for the topics `tgif`, `wecandateif`, `ifwedate`, `MentionSomeoneHandsome`, and `mentionsomeonebeatiful` respectively from top to bottom. From the top three topics (“`tgif`,” “`wecandateif`,” and “`ifwedate`”) in Fig.4.14, we find that malicious account peaks tend to follow authenticated account peaks. This observation is likely to reveal one strategy of malicious accounts: focusing on those topics that have high trending potential right before they go trending. In the meantime, we can see that malicious account peaks and authenticated account peaks interweave to make the topics trend from the last two topics (“`MentionSomeoneHandsome`” and “`mentionsomeonebeatiful`”) in Fig.4.14. A possible explanation is that these authenticated accounts happen to synchronize with malicious accounts to make the topics trend. In other words, the strategies of making the topics trend

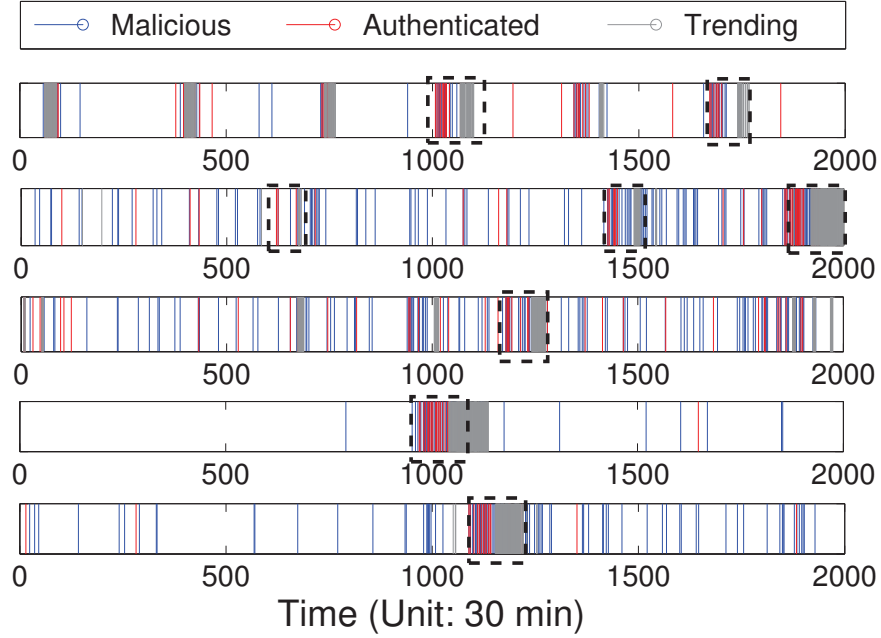


Figure 4.14: Malicious and Authenticated Account Peaks for the Topics

include the involvement of authenticated accounts and spamming tactics.

4.5 Discussion on Manipulation of Trends

Spammers in Twitter conduct malicious activities mainly through compromised and sybil accounts. In this section, we first evidence the involvement of compromised and sybil accounts in the manipulation of trends, and then we simulate the manipulation of dynamics as compromised and sybil accounts would do. Finally, we discuss the possible countermeasures against the manipulation of trends.

4.5.1 Compromised Accounts

Account compromise enables spammers to hijack followers and tweet history immediately. Therefore, compromised accounts are very likely to be employed for manipulating the trends.

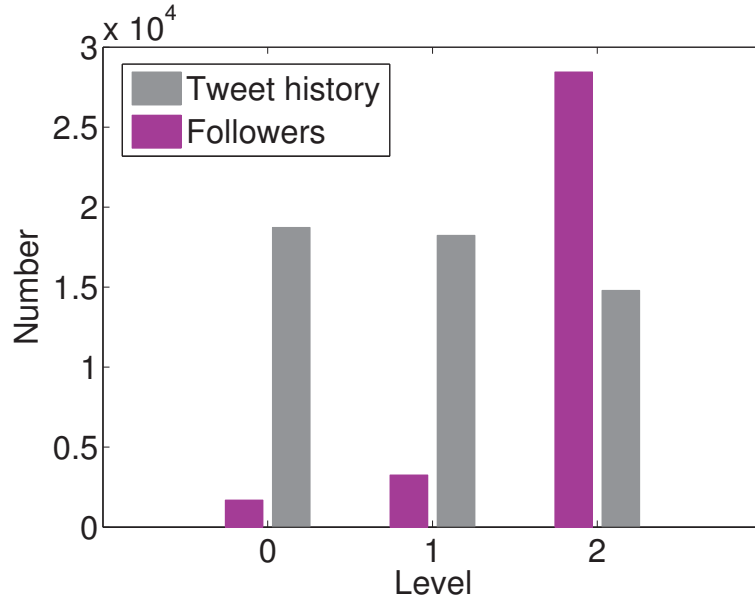


Figure 4.15: Avg Follower and Tweet History Number of Spammer and Descendants

We then examine the follower number and tweet history for the identified spammers (level 0), as well as the first and second generations of their descendants (levels 1 and 2). Fig.4.15 depicts the average follower number and tweet history for the spammers and their descendants. We observe that as the level increases, the average follower number increases exponentially while the average tweet history decreases. The mostly likely explanation is that there exist compromised accounts in the followers of the identified spammers. Spammers use the compromised accounts to increase the follower number for a topic and thereby increase the topic’s credibility. Thus, the possibility of the topic trending can be significantly increased. Meanwhile, the compromised accounts do not need to be very active, but spammers could manipulate the tweet history of a topic by performing frequent activities.

Therefore, compromised accounts pose a serious threat to the security of Twitter trends in that they can be used to manipulate the follower dynamics and tweet history dynamics.

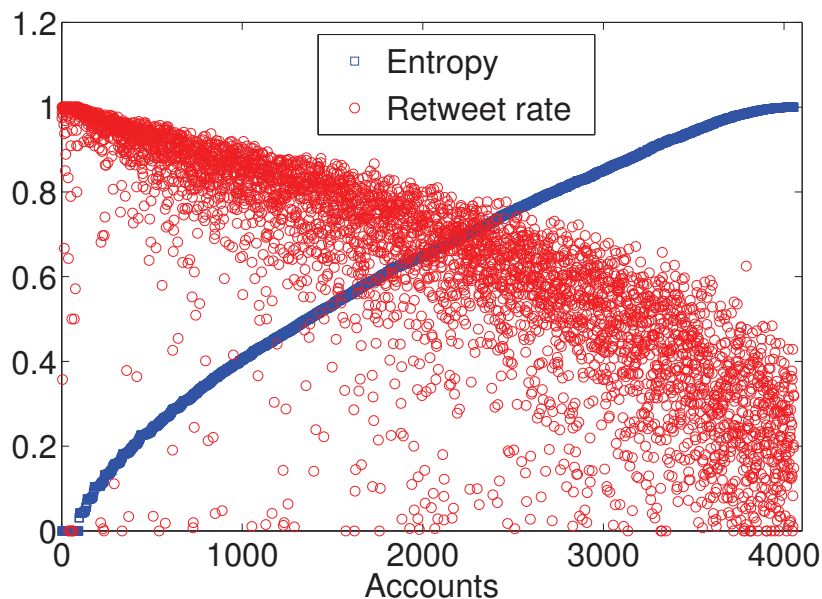


Figure 4.16: Entropy and Retweet Rate of the Accounts

4.5.2 Sybil Accounts

According to a recent report [23], an enormous number of sybil accounts on Twitter are run by bot-masters. They are sold and bought through an underground market [94]. To verify the existence of sybil accounts in the manipulation of Twitter trending, we study the behavior profile of those accounts that appear in the spike, in which the evidence of trending manipulation is found. There are of total 4,055 accounts (5,193 tweets) in the spike. Using the web crawling method, we extract a collection of tweets posted by each account and the related information (e.g., follower number) for each account. There are on average 180 tweets for one account, and the tweet histories last 334 days on average. We first explore the entropy of time intervals between posting tweets for each account. The entropy of time intervals between posting tweets of an account can indicate the regularity of the account’s posting behavior. In general, the smaller entropy value an account has, the more likely it is a bot. Fig.4.16 shows the entropy (ascending order) of the accounts.

At the same time, sybil accounts are not likely to have their own opinion. Therefore, they generally do not post original tweets but tend to retweet. We also calculate the retweet rate of the accounts mentioned above and illustrate the result in Fig.4.16. It is observed that entropy is inversely proportional to the overall retweet rate. There exist some accounts that have considerably low entropy but a high retweet rate. In other words, they regularly retweet the posts from others and rarely post original tweets. Although we are not going to single out individual sybil accounts, the posting behaviors of the accounts above is the same as (or very close to) those of sybil accounts.

To further confirm our conjecture, we compare the ratio of friend to follower number between the top 10% accounts (with lower entropy and higher retweet rate) and all accounts in Fig.4.16. If account A follows account B , A is B 's follower, and B is A 's friend. The intuition is that sybil accounts have no personal opinion and hence they generally cannot attract many followers. Fig. 4.17 illustrates the CDF of the ratio of friend number to the follower number for the top 10% accounts and that for all accounts. We can see that the top 10% accounts have a larger ratio of friend number to the follower number than all accounts on average. It supports our conjecture on the active involvement of sybil accounts in the manipulation of Twitter trending.

4.5.3 The Manipulation of Dynamics

It is straightforward for the trending algorithm of Twitter to emphasize the dynamics of a topic. We examine whether compromised and sybil accounts manipulate the trends by manipulating the dynamics. As discussed above, compromised and sybil accounts can significantly impact tweet dynamics, account dynamics, follower dynam-

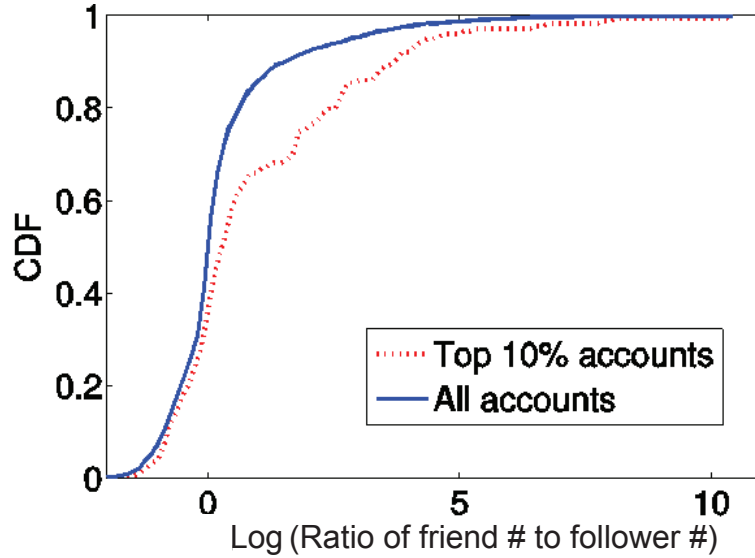


Figure 4.17: Ratio of Friend Number to Follower Number

ics, and tweet history dynamics. To quantify the manipulation through the dynamics, we conduct a simulation on the manipulation of dynamics. To do this, we locate the peaks of the dynamics and sum the adjacent peaks into one peak. The intuition is that each peak in the dynamics is likely to represent an effort of spammers to produce a trend. Therefore, if multiple peaks of the dynamics could be summed into one, it is more likely to produce a trend. We simulate the manipulation of the dynamics by summing two and three adjacent peaks. Then the SVM classifier is employed to predict how many times of trends the manipulated dynamics will produce than the original dynamics. Fig.4.18 shows the results averaged over all the manipulated dynamics. Both manipulated dynamics well outperform the original dynamics in terms of the possibility of producing trends. Consequently, it further indicates the threat from compromised and sybil accounts for manipulating the trends.

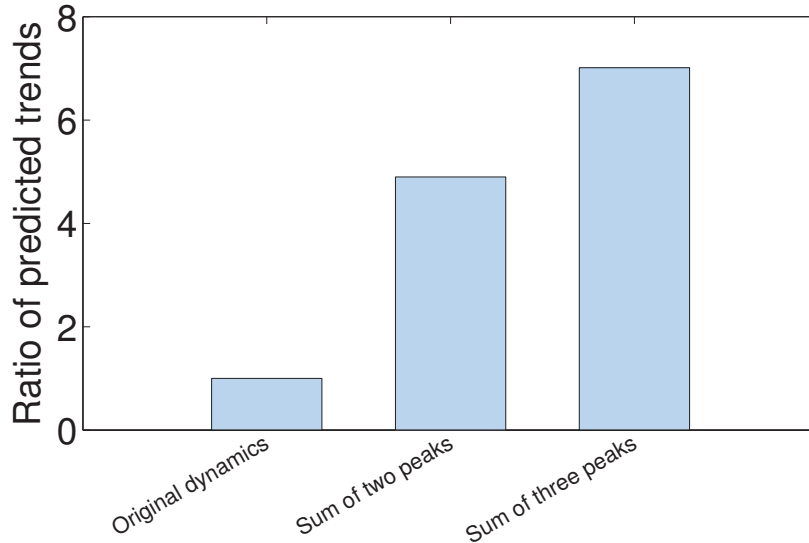


Figure 4.18: Ratio of Predicted Trends

4.5.4 Countermeasures Against Trending Manipulation

We briefly describe three different ways to defend against trending manipulation in Twitter, and we will explore more effective defense in our future work.

Strengthening the Twitter Trending Algorithm The detailed Twitter trending algorithm remains unknown. Meanwhile, due to the limitations of the dataset, we study only the simple factors of Twitter trending (i.e., tweet number). However, using the evidence of manipulation we demonstrated before, we believe the algorithm of Twitter trending can be strengthened by considering more complicated factors. For example, network characteristics can be taken into consideration, such as cliques. *Cliques* represent the dense clusters in graphs [25]. In general, complete cliques tend to represent interesting topics. Although spammers could produce cliques, it will no doubt increase their risk of being suspended.

Detecting the Real-time Anomalies of Twitter Trending Due to the outbreak nature of Twitter trends, we need to detect the anomalies of Twitter trending in real-time. Regarding that trends are usually manipulated by compromised and

fake accounts which are in hand of a few malicious users, we can detect the anomaly of tweet source as an indicator of trend manipulation. Moreover, the monitoring of *topological hierarchy* of the accounts in a topic help detect trend anomaly. As figure 4.6 shows, spamming infrastructure exists in the topological hierarchy of spam accounts and this kind of anomaly indicates trend manipulation.

Detecting Manipulation Using Historical Manipulated Topics We can classify different topics into two classes: manipulated and normal. There should be some connections among manipulated topics due to similar manipulation strategies. The connections among normal trending topics and the connections among manipulated topics, can be exploited for the early detection of Twitter trends using previously trending topics [78]. One feasible way to trace the connection between two topics with respect to manipulation is to treat one topic as the training set and the other as the testing set. In this regard, an SVM classifier can be employed to train the classification model based on the training set and then perform the classification task based on the testing set. The classification result reflects the connection between the two topics. Thus, the connections among manipulated topics enable us to detect manipulated topics one by one from the very beginning of identifying the first set of manipulated topics. The challenges here include identifying the first set of manipulated topics and verifying the manipulated topics. The influence model that we use to demonstrate the evidence of manipulation can be utilized to identify the first set of manipulated topics. The development of an accurate and practical verification method remains as our future work.

4.5.5 Limitation and Future Work

There are some limitations of our work and some remain as future work.

First, we use a linear influence model to capture the network effect on the diffusion of a topic in Twitter, which enable us to find the evidence of manipulation. The model applies to linear scenarios and to develop a non-linear model remains one of the future work.

Second, we randomly choose 11 topics and more than 10,000 related tweets to infer the relevance of 5 key factors over Twitter trending. Although we try our best to guarantee the randomness, those 11 sample topics may not be large enough to represent the overall scenario in practice. Besides, we study 5 comparatively straightforward factors that may affect trending. In the future work, we plan to consider more complicated factors and sample more topics to study the factors over trending.

Finally, we discuss the countermeasures against Twitter trends manipulation but most of them remain in the discussion stage. We leave the implementation and evaluation of those countermeasures as our future work. Specifically, we plan to develop a manipulation detection mechanism by using an SVM classifier. We will train the classifier using previously manipulated topics and classify future trends as manipulated or not.

4.6 Related Works

To the best of our knowledge, this is the first effort to investigate whether Twitter trends could be manipulated.

Research on trending topics in Twitter includes real-world event recognition [34, 113], realtime trending topic detection [25, 38, 59, 69], the evolution of trending topic characterization [27, 28], and the taxonomy of trending topics [55, 63, 76]. Becker *et al.* [34] analyzed the stream of Twitter messages and distinguished the messages about real-world events from non-event messages based on a clustering

method. Zubiaga *et al.* [113] categorized different triggers that leverage the trending topics by using social features rather than content-based approaches.

In the detection of realtime trending topics, Agarwal *et al.* [25] identified the emerging events before they became trending topics by modeling the detection problem as discovering dense clusters in highly dynamic graphs. Kasiviswanathan *et al.* [59] presented a dictionary-learning-based framework for detecting emerging topics in social media via the user-generated stream. Lu *et al.* [69] used an energy function to model the life activity of news events on Twitter and proposed a news event detection method based on online energy function. Cataldi *et al.* [38] identified emerging terms from user content by measuring user authority and proposing a keyword life cycle model, and then detected the emerging topics by formalizing the keyword-based topic graph.

To address the evolution and taxonomy of trending topics, Altshuler and Pan [27] presented the lower bounds of the probability that emerging trends successfully spread through the scale-free networks. Asur *et al.* [28] studied trending topics on Twitter and theoretically analyzed the formation, persistence, and decay of trends. Naaman *et al.* [76] characterized the trends in multiple dimensions and presented a taxonomy of trends. They also proposed a collection of hypotheses on different kinds of trends and evaluated them. Lehmann *et al.* [63] classified the popular hashtags by the temporal dynamics of hashtags. Irani *et al.* [55] focused on the trend-stuffing issue and developed a classifier to automatically identify the trend-stuffing in tweets.

Whether a topic begins trending is closely related to (1) the influence of users who are involved with the topic and (2) the topic adoption for users who are exposed to the topic. Cha *et al.* [39] performed a comparison of three different measures of influence: indegree, retweet, and mention. Weng *et al.* [99] proposed a topic-sensitive PageRank measure for user influence. Romero *et al.* [84] proposed an algorithm

to measure the relative influence and passivity of each user from the viewpoint of a whole network. Bakshy *et al.* [32] measured the influence from the diffusion tree. The studies of topic adoption in Twitter mainly concentrate on hashtag adoption. Lin *et al.* [67] classified the adoption of hashtags into two classes and proposed a framework to capture the dynamics of hashtags based on their topicality, interactivity, diversity, and prominence. Yang *et al.* [107] studied the effect of the dual role of a hashtag on hashtag adoption.

4.7 Summary

With the datasets we collected via Twitter API, we first evidence the manipulation of Twitter trending and observe a suspect spamming infrastructure. Then, we employ the SVM classifier to explore how accurately five different factors at the topic level (popularity, coverage, transmission, potential coverage, and reputation) could predict the trending. We observe that, except for transmission, the other factors are all closely related to Twitter trending. We further investigate the interacting patterns between authenticated accounts and malicious accounts. Finally, we present the threat posed by compromised and sybil accounts to Twitter trending and discuss the corresponding countermeasures against trending manipulation.

Conclusion

In this dissertation, we first examine the effectiveness of OSN privacy settings for protecting user privacy. Given each privacy configuration, we propose a corresponding scheme to reveal the target user’s basic profile and connection information starting from some leaked connections on its homepage. Based on the dataset we collect on Facebook, we derive the privacy exposure in each privacy setting type and measure the accuracy of our privacy inference schemes given different amount of public information. The evaluation results show that a user’s basic private profile can be inferred with high accuracy and connections can be revealed in a significant portion based on even a small number of directly leaked connections.

We then propose a behavioral-profile-based method to detect OSN user account compromise in a timely manner. Specifically, we propose eight behavioral features to portray a user’s social behavior. A user’s statistical distributions of those feature values comprise its behavioral profile. Based on the sample data we collected from Facebook, we find that each user’s activities highly conform to its behavioral profile while different users’ profile tend to diverge from each other, which can be employed for compromise detection. The evaluation results show that the more complete and accurate a user’s behavioral profile can be built, the more accurate

compromisation detection can be achieved.

Finally, we investigate the manipulation of OSN trending topics. Based on the dataset we collected from Twitter, we manifest the manipulation of trending and a suspect spamming infrastructure. We then measure how accurately the five factors (popularity, coverage, transmission, potential coverage, and reputation) can predict trending using an SVM classifier. We further study the interaction patterns between authenticated accounts and malicious accounts in trending. At the end, we demonstrate the threat of compromised accounts and sybil accounts to trending using simulation and discuss countermeasures against trending manipulation.

References

- [1] 250,000 twitter accounts hacked. <http://www.cnn.com/2013/02/01/tech/social-media/twitter-hacked>.
- [2] 50,000 facebook accounts hacked. <http://www.ktsm.com/news/thousands-of-facebook-accounts-hacked>.
- [3] Detecting suspicious account activity. <http://googleonlinesecurity.blogspot.com/2010/03/detecting-suspicious-account-activity.html>.
- [4] Evolving Roles of News on Twitter and Facebook. <http://www.journalism.org/2015/07/14/the-evolving-role-of-news-on-twitter-and-facebook/>.
- [5] Facebook Login Approval. <https://www.facebook.com/help/www/163190627080285>.
- [6] Facebook name policy. <http://www.facebook.com/help/?page=258984010787183>.
- [7] Facebook newsroom. <http://newsroom.fb.com/>.
- [8] Facebook passes 1.65 billion monthly active users. <http://venturebeat.com/2016/04/27/facebook-passes-1-65-billion-monthly-active-users-54-access-the-service-only-on-mobile/>.
- [9] Facebook tracks the location of logins for better security. <http://www.zdnet.com/blog/weblife/facebook-adds-better-security-tracks-the-location-of-your-logins/2010>.
- [10] Google trend manipulation. <http://piloseo.com/google/trends-manipulation/>.

- [11] How Facebook Uses Your Data to Target Ads, Even Offline. <http://lifehacker.com/5994380/how-facebook-uses-your-data-to-target-ads-even-offline>.
- [12] IGRAPH. <http://igraph.sourceforge.net/>.
- [13] Introducing Login Approvals. https://www.facebook.com/note.php?note_id=10150172618258920.
- [14] Kalman filter,. http://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf.
- [15] New Tools to Secure a Compromised Account. <https://blog.facebook.com/blog.php?post=107720572130>.
- [16] Social media spam increased 355% in first half of 2013. <http://mashable.com/2013/09/30/social-media-spam-study/>.
- [17] Social Media Survey: Privacy, Security Concerns Persist. http://www.informationweek.com/social-business/social_networking_consumer/social-media-survey-privacy-security-con/232600668.
- [18] Social Media to Lead Parade of IPOs in year ahead. http://www.nbcnews.com/id/41179092/ns/business-stocks_and_economy/t/social-media-lead-parade-ipos-year-ahead/.
- [19] Statistical learning theory. Wiley, 1998.
- [20] There are more than 200M monthly active twitter users. <https://twitter.com/twitter/status/281051652235087872>.
- [21] Throughback Thursday. <http://time.com/3707773/instagram-throwbackthursday/>.
- [22] Twitter account of Thomson Reuters hacked by Syrian activists. <http://www.nbcnews.com/technology/twitter-account-thomson-reuters-hacked-syrian-activists-6C10790501>.
- [23] Wall street journal (inside a twitter robot factory). <http://www.wsj.com/articles/SB10001424052702304607104579212122084821400>.

- [24] Zuckerberg’s Facebook page hacked to prove security flaw. <http://www.cnn.com/2013/08/19/tech/social-media/zuckerberg-facebook-hack/index.html>.
- [25] M. K. Agarwal, K. Ramamritham, and M. Bhide. Real time discovery of dense clusters in highly dynamic graphs: Identifying real world events in highly dynamic environments. In *Proceedings of the VLDB Endowment 2012*, PVLDB’12, pages 980–991. ACM, 2012.
- [26] R. Agrawal, M. Potamias, and E. Terzi. Learning the nature of information in social networks. In *Sixth International AAAI Conference on Weblogs and Social Media*, ICWSM’12, Dublin, Ireland, 2012. The AAAI Press.
- [27] Y. Altshuler, W. Pan, and A. S. Pentland. Trends prediction using social diffusion models. In *Proceedings of the 5th International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction*, SBP’12, pages 97–104, College Park, MD, 2012. Springer-Verlag.
- [28] S. Asur, B. a. Huberman, G. Szabo, and C. Wang. Trends in social media: Persistence and decay. In *Fifth International conference on weblogs and social media*, ICWSM’11, Barcelona, Spain, 2011.
- [29] Y. Bachrach, M. Kosinski, T. Graepel, P. Kohli, and D. Stillwell. Personality and patterns of facebook usage. In *Proceedings of the 3rd Annual ACM Web Science Conference*, WebSci’12, pages 24–32, Evanston, Illinois, USA, 2012. ACM.
- [30] L. Backstrom, C. Dwork, and J. Kleinberg. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *Proceedings of the 16th WWW’07*, 2007.
- [31] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin. Persona: an online social network with user-defined privacy. In *Proceedings of the 2009 ACM SIGCOMM*, 2009.
- [32] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts. Everyone’s an influencer: quantifying influence on twitter. In *The Fourth ACM International Con-*

- ference on Web Search and Data Mining, WSDM'11*, pages 65–74, HongKong, China, 2011. ACM.
- [33] M. Balduzzi, C. Platzer, T. Holz, E. Kirda, D. Balzarotti, and C. Kruegel. Abusing social networks for automated user profiling. In *Proceedings of the 13th RAID'10*, 2010.
- [34] H. Becker, M. Naaman, and L. Gravano. Beyond trending topics: Real-world event identification on twitter. In *Fifth International AAAI Conference on Weblogs and Social Media, ICWSM'11*, Barcelona, Spain, 2011.
- [35] F. Benevenuto, T. Rodrigues, M. Cha, and V. Almeida. Characterizing user behavior in online social networks. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference, IMC'09*, pages 49–62, Chicago, Illinois, USA, 2009. ACM.
- [36] J. Bonneau, J. Anderson, R. Anderson, and F. Stajano. Eight friends are enough: social graph approximation via public listings. In *Proceedings of the 2nd ACM EuroSys Workshop on SNS'09*, 2009.
- [37] Q. Cao, M. Sirivianos, X. Yang, and T. Pregueiro. Aiding the detection of fake accounts in large scale social online services. In *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation, NSDI'12*, San Jose, CA, 2012. USENIX Association.
- [38] M. Cataldi, L. Di Caro, and C. Schifanella. Emerging topic detection on twitter based on temporal and social terms evaluation. In *Proceedings of the Tenth International Workshop on Multimedia Data Mining, MDMKDD'10*, pages 4:1–4:10, Washington, D.C., 2010. ACM.
- [39] M. Cha, H. Haddadi, F. Benevenuto, and K. Gummadi. Measuring user influence in twitter: The million follower fallacy. In *4th Int'l AAAI Conference on Weblogs and Social Media, ICWSM'10*, Washington D.C., USA, 2010.
- [40] A. Chaabane, G. Acs, and M. A. Kaafar. You are what you like! information leakage through users' interests. In *Proceedings of the 19th NDSS'12*, 2012.

- [41] C. C. Chang and C. J. Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, pages 27:1–27:27, 2011.
- [42] T. M. Cover and J. A. Thomas. *Elements of Information Theory, 2nd Edition*. Wiley-Interscience, 2006.
- [43] G. Danezis and P. Mittal. Sybilinfer: Detecting sybil nodes using social networks. In *Proceedings of the 16th Network and Distributed System Security Symposium, NDSS'09*, San Diego, California, USA, 2009.
- [44] M. Egele, G. Stringhini, C. Kruegel, and G. Vigna. Compa: Detecting compromised accounts on social networks. In *Symposium on Network and Distributed System Security, NDSS'13*, San Diego, CA USA. Internet Society.
- [45] R. Eyal, S. Kraus, and A. Rosenfeld. Identifying missing node information in social networks. *Artificial Intelligence*, pages 1166–1172, 2011.
- [46] A. J. Feldman, A. Blankstein, M. J. Freedman, and E. W. Felten. Social networking with frientegrity: Privacy and integrity with an untrusted provider. In *the 21st USENIX Security'12*, Aug 2012.
- [47] H. Gao, Y. Chen, and K. Lee. Towards online spam filtering in social networks. In *Symposium on Network and Distributed System Security, NDSS'12*, San Diego, CA USA. Internet Society.
- [48] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Y. Zhao. Detecting and characterizing social spam campaigns. In *Proceedings of the 10th IMC'10*, 2010.
- [49] J. Ginsberg, M. H. Mohebbi, R. S. Patel, L. Brammer, M. S. Smolinski, and L. Brilliant. Detecting influenza epidemics using search engine query data. *Nature* 457, pages 1012–1014, 2009.
- [50] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.
- [51] K.-I. Goh and A.-L. Barabási. Burstiness and memory in complex systems. *EPL (Europhysics Letters)*, 81(4):48002, 2008.

- [52] C. Grier, K. Thomas, V. Paxson, and M. Zhang. @spam: the underground on 140 characters or less. In *Proceedings of the 17th ACM conference on Computer and communications security*, CCS'10, pages 27–37, Chicago, Illinois, USA, 2010. ACM.
- [53] P. Gundecha, G. Barbier, and H. Liu. Exploiting vulnerability to secure user privacy on a social networking site. In *Proceedings of the 17th ACM KDD'11*, 2011.
- [54] S. Hao, N. A. Syed, N. Feamster, A. G. Gray, and S. Krasser. Detecting spammers with snare: spatio-temporal network-level automatic reputation engine. In *Proceedings of the 18th conference on USENIX security symposium*, SSYM'09, pages 101–118, Berkeley, CA, USA, 2009. USENIX Association.
- [55] D. Irani, S. Webb, C. Pu, F. Drive, and B. Gsrc. Study of trend-stuffing on twitter through text classification. In *2010 Collaboration, Electronic messaging, Anti-Abuse and Spam Conference*, CEAS'10, Seattle, WA, USA, 2010.
- [56] G. Jacob, E. Kirda, C. Kruegel, and G. Vigna. Pubcrawl: protecting users and businesses from crawlers. In *Proceedings of the 21st USENIX conference on Security symposium*, Security'12, pages 25–25, Bellevue, WA, 2012. USENIX Association.
- [57] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*, ECML'98, pages 137–142, Chemnitz, Germany,, 1998. Springer-Verlag.
- [58] M. Just, A. Crigler, P. Metaxas, and E. Mustafaraj. It's trending on twitter-an analysis of the twitter manipulations in the massachusetts 2010 special senate election. In *APSA 2012 Annual Meeting Paper*, 2012.
- [59] S. P. Kasiviswanathan, P. Melville, A. Banerjee, and V. Sindhvani. Emerging topic detection using dictionary learning. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM'11, pages 745–754, Glasgow, Scotland, UK, 2011. ACM.

- [60] A. Korolova, R. Motwani, S. U. Nabar, and Y. Xu. Link privacy in social networks. In *Proceedings of the 17th ACM CIKM'08*, 2008.
- [61] K. Lee, J. Caverlee, and S. Webb. Uncovering social spammers: social honeypots + machine learning. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR'10, pages 435–442, Geneva, Switzerland, 2010. ACM.
- [62] P. Lee, K., R. D., Narayanan, M. M. A. Patwary, A. Agrawal, and A. Choudhary. Twitter trending topic classification. In *IEEE 11th International Conference on Data Mining Workshops*, ICDMW'11, pages 251–258, Washington, DC, USA, 2011. IEEE Computer Society.
- [63] J. Lehmann, B. Goncalves, J. J. Ramasco, and C. Cattuto. Dynamical classes of collective attention in twitter. In *Proceedings of the 21st International Conference on World Wide Web*, WWW'12, pages 251–260, Lyon, France, 2012. ACM.
- [64] V. Leroy, B. B. Cambazoglu, and F. Bonchi. Cold start link prediction. In *Proceedings of the 16th ACM KDD'10*, 2010.
- [65] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Proceedings of the 12th CIKM'03*, 2003.
- [66] J. Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, Sept. 2006.
- [67] Y. R. Lin, D. Margolin, B. Keegan, A. Baronchelli, and D. Lazer. Bigbirds never die: Understanding social dynamics of emergent hashtag. In *Proceedings of the 7th International AAAI Conference on Weblogs and Social Media*, ICWSM'13, Cambridge, MA, USA, 2013. The AAAI Press.
- [68] Y. Liu, K. P. Gummadi, B. Krishnamurthy, and A. Mislove. Analyzing facebook privacy settings: user expectations vs. reality. In *Proceedings of the 2011 ACM SIGCOMM IMC'11*, 2011.
- [69] R. Lu, Z. Xu, Y. Zhang, and Q. Yang. Life activity modeling of news event. In *Proceedings of the 16th Pacific-Asia Conference on PAKDD*, PAKDD'12, pages 73–84, Kuala Lumpur, Malaysia, 2012. Springer Berlin Heidelberg.

- [70] M. Madejski, M. Johnson, and S. M. Bellovin. A study of privacy setting errors in an online social network. In *Proceedings of SESOC'12*, 2012.
- [71] D. Mashima, P. Sarkar, E. Shi, C. Li, R. Chow, and D. Song. Privacy settings from contextual attributes: A case study using google buzz. In *PerCom Workshops*, pages 257–262. IEEE, 2011.
- [72] A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel. You are who you know: inferring user profiles in online social networks. In *Proceedings of the 3rd ACM WSDM'10*, 2010.
- [73] M. Mondal, B. Viswanath, A. Clement, P. Druschel, K. P. Gummadi, A. Mislove, and A. Post. Limiting large-scale crawls of social networking sites. *SIGCOMM Computer Communication Review*, 41(4):398–399, 2011.
- [74] M. Mondal, B. Viswanath, A. Clement, P. Druschel, K. P. Gummadi, A. Mislove, and A. Post. Defending against large-scale crawls in online social networks. In *Proceeding of the 8th International Conference on emerging Networking Experiments and Technologies*, CoNext'12, pages 325–336, Nice, France, 2012. ACM.
- [75] F. Morstatter, J. Pfeffer, H. Liu, and K. M. Carley. Is the sample good enough? comparing data from twitter's streaming api with twitter's firehose. *The 7th International AAAI Conference on Weblogs and Social Media*, abs/1306.5204, 2013.
- [76] M. Naaman, H. Becker, and L. Gravano. Hip and trendy: Characterizing emerging trends on twitter. *Journal of the American Society for Information Science and Technology*, pages 902–918, 2011.
- [77] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *Proceedings of 30th IEEE Symposium on Security and Privacy (S&P'09)*, may 2009.
- [78] S. Nikolov. Trend or no trend: A novel nonparametric method for classifying time series. *Doctoral dissertation*, 2012.
- [79] C. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Proceedings of the International Conference on Computer Vision, 1998*, pages 555–562, Bombay, India. IEEE.

- [80] P. Pedarsani and M. Grossglauser. On the privacy of anonymized networks. In *Proceedings of the 17th ACM KDD'11*, 2011.
- [81] I. Polakis, M. Lancini, G. Kontaxis, F. Maggi, S. Ioannidis, A. D. Keromytis, and S. Zanero. All your face are belong to us: breaking facebook's social authentication. In *Proceedings of the 28th Annual Computer Security Applications Conference, ACSAC'12*, pages 399–408, Orlando, Florida, 2012. ACM.
- [82] A. Post, V. Shah, and A. Mislove. Bazaar: strengthening user reputations in on-line marketplaces. In *Proceedings of the 8th USENIX conference on Networked systems design and implementation, NSDI'11*, pages 14–14, Boston, MA, 2011. USENIX Association.
- [83] J. Ratkiewicz, M. Conover, M. Meiss, B. Gonçalves, S. Patil, A. Flammini, and F. Menczer. Truthy: Mapping the spread of astroturf in microblog streams. In *Proceedings of the 20th International Conference Companion on World Wide Web, WWW'11*, pages 249–252, Hyderabad, India, 2011.
- [84] D. M. Romero, W. Galuba, S. Asur, and B. A. Huberman. Influence and passivity in social media. In *Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases, ECML PKDD'11*, pages 18–33, Athens, Greece, 2011. Springer-Verlag Berlin Heidelberg.
- [85] C. Ross, E. S. Orr, M. Sisic, J. M. Arseneault, M. G. Simmering, and R. R. Orr. Personality and motivations associated with facebook use. *Comput. Hum. Behav.*, pages 578–586, 2009.
- [86] F. Schneider, A. Feldmann, B. Krishnamurthy, and W. Willinger. Understanding online social network usage from a network perspective. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference, IMC'09*, pages 35–48, Chicago, Illinois, USA, 2009. ACM.
- [87] K. Singh, S. Bhola, and W. Lee. xbook: redesigning privacy control in social networking platforms. In *Proceedings of the 18th USENIX security symposium, SSYM'09*, Berkeley, CA, USA, 2009. USENIX Association.
- [88] J. Song, S. Lee, and J. Kim. Spam filtering in twitter using sender-receiver relationship. In *Proceedings of the 14th international conference on Recent*

Advances in Intrusion Detection, RAID'11, pages 301–317, Menlo Park, CA, 2011. Springer-Verlag.

- [89] J. Staddon. Finding "hidden" connections on linkedin an argument for more pragmatic social network privacy. In *Proceedings of the 2nd ACM workshop AISec'09*, 2009.
- [90] G. Stringhini, C. Kruegel, and G. Vigna. Detecting spammers on social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference*, ACSAC'10, pages 1–9, Austin, Texas, 2010. ACM.
- [91] C.-H. Tai, P. S. Yu, D.-N. Yang, and M.-S. Chen. Privacy-preserving social network publication against friendship attacks. In *Proceedings of the 17th ACM KDD'11*, 2011.
- [92] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song. Design and evaluation of a real-time url spam filtering service. In *IEEE Symposium on Security and Privacy*, S&P'11, pages 447–462. IEEE Computer Society, 2011.
- [93] K. Thomas, C. Grier, and V. Paxson. Adapting social spam infrastructure for political censorship. In *Proceedings of the 5th USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, the 5th USENIX LEET'12, April 2012.
- [94] K. Thomas, D. McCoy, C. Grier, A. Kolcz, and V. Paxson. Trafficking fraudulent accounts: The role of the underground market in twitter spam and abuse. In *Proceedings of 22nd USENIX Security Symposium*, USENIX Security'13, 2013.
- [95] S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *Proceedings of the Ninth ACM International Conference on Multimedia*, MULTIMEDIA'01, pages 107–118, Ottawa, Canada, 2001. ACM.
- [96] B. Viswanath, A. Post, K. P. Gummadi, and A. Mislove. An analysis of social network-based sybil defenses. In *Proceedings of the 2010 ACM SIGCOMM'10*, 2010.
- [97] G. Wang, T. Konolige, C. Wilson, X. Wang, H. Zheng, and B. Y. Zhao. You are how you click: Clickstream analysis for sybil detection. In *Proceedings*

of *22nd USENIX Security Symposium*, USENIX Security'13, pages 241–256, Washington D.C., USA, 2013.

- [98] G. Wang, M. Mohanlal, C. Wilson, X. Wang, M. J. Metzger, H. Zheng, and B. Y. Zhao. Social turing tests: Crowdsourcing sybil detection. In *Proceedings of the 20th Network and Distributed System Security Symposium*, NDSS'13, San Diego, California, USA, 2013. The Internet Society.
- [99] J. Weng, E.-P. Lim, J. Jiang, and Q. He. Twiterrank: Finding topic-sensitive influential twitterers. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM'10, pages 261–270, New York, New York, USA, 2010. ACM.
- [100] C. Wilson, B. Boe, A. Sala, K. P. Puttaswamy, and B. Y. Zhao. User interactions in social networks and their implications. In *Proceedings of the 4th ACM European conference on Computer systems*, EuroSys'09, pages 205–218, Nuremberg, Germany, 2009. ACM.
- [101] G. Wondracek, T. Holz, E. Kirda, and C. Kruegel. A practical attack to de-anonymize social network users. In *Proceedings of the 2010 IEEE Symposium on Security and Privacy (S&P'10)*, 2010.
- [102] Y. Xie, F. Yu, K. Achan, R. Panigrahy, G. Hulten, and I. Osipkov. Spamming botnets: signatures and characteristics. In *Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication*, SIGCOMM'08, pages 171–182, Seattle, WA, USA, 2008. ACM.
- [103] Y. Xie, F. Yu, Q. Ke, M. Abadi, E. Gillum, K. Vitaldevaria, J. Walter, J. Huang, and Z. M. Mao. Innocent by association: early recognition of legitimate users. In *ACM Conference on Computer and Communications Security*, CCS'12, pages 353–364, Raleigh, NC, USA, 2012. ACM.
- [104] C. Yang, R. Harkreader, J. Zhang, S. Shin, and G. Gu. Analyzing spammers' social networks for fun and profit: a case study of cyber criminal ecosystem on twitter. In *Proceedings of the 21st international conference on World Wide Web*, WWW'12, pages 71–80, Lyon, France, 2012. ACM.

- [105] C. Yang, R. C. Harkreader, and G. Gu. Die free or live hard? empirical evaluation and new design for fighting evolving twitter spammers. In *Proceedings of the 14th international conference on Recent Advances in Intrusion Detection, RAID'11*, pages 318–337, Menlo Park, CA, 2011. Springer-Verlag.
- [106] J. Yang and J. Leskovec. Modeling information diffusion in implicit networks. In *Proceedings of the 10th IEEE International Conference on Data Mining, ICDM'10*, Sydney, Australia. IEEE Computer Society.
- [107] L. Yang, T. Sun, M. Zhang, and Q. Mei. We know what @you #tag: Does the dual role affect hashtag adoption? In *Proceedings of the 21st International Conference on World Wide Web, WWW'12*, pages 261–270, Lyon, France, 2012. ACM.
- [108] Y. Yang, J. Lutes, F. Li, B. Luo, and P. Liu. Stalking online: on user privacy in social networks. In *Proceedings of the second ACM CODASPY'12*, New York, NY, USA, 2012.
- [109] X. Ying and X. Wu. On link privacy in randomizing social networks. In *Proceedings of the 13th PAKDD'09*, 2009.
- [110] X. Zhao, L. Li, and G. Xue. Authenticating strangers in online social networks. *International Journal of Security and Networks*, 6(4):237–248, Jan. 2011.
- [111] Y. Zhao, Y. Xie, F. Yu, Q. Ke, Y. Yu, Y. Chen, and E. Gillum. Botgraph: large scale spamming botnet detection. In *Proceedings of the 6th USENIX symposium on Networked systems design and implementation, NSDI'09*, pages 321–334, Boston, MA, USA, 2009. USENIX Association.
- [112] E. Zheleva and L. Getoor. To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles. In *Proceedings of the 18th WWW'09*, 2009.
- [113] A. Zubiaga, D. Spina, and R. Martinez. Classifying trending topics: A typology of conversation triggers on twitter. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management, CIKM'11*, pages 2461–2464, Glasgow, Scotland, UK, 2011. ACM.

VITA

Xin Ruan was born in Xi'an, Shaanxi Province, China, on September 29, 1985. She received her Bachelor's degree in Computer Science in 2007 from Xidian University in Xi'an and also received her Master of Science degree in Computer Science there in 2009. She came to the Department of Computer Science at the College of William and Mary for a Ph.D. degree in 2009.

This dissertation was defended on October 10, 2016 at the College of William and Mary in Virginia.