2015

Detect Spammers in Online Social Networks

Yi Zhang

Detect Spammers in Online Social Networks

By

Yi Zhang

A Thesis
Submitted to the Faculty of Graduate Studies
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science
at the University of Windsor

Windsor, Ontario, Canada

2015

Detect Spammers in Online Social Networks

by

Yi Zhang

APPROVED BY:

A. Hussein
Department of Mathematics and Statistics

A. Jaekel
School of Computer Science

J. Lu, Advisor
School of Computer Science

January 27, 2015

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyones copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

Fake followers in online social networks (OSNs) are the accounts that are created to boost the rank of some targets. These spammers can be generated by programs or human beings, making them hard to identify. In this thesis, we propose a novel spammer detection method by detecting near-duplicate accounts who share most of the followers.

It is hard to discover such near-duplicates on large social networks that provide limited remote access. We identify the near-duplicates and the corresponding spammers by estimating the Jaccard similarity using star sampling, a combination of uniform random sampling and breadth-first crawling. Then we applied our methods in Sina Weibo and Twitter. For Weibo, we find 395 near-duplicates, 12 millions suspected spammers and 741 millions spam links. In Twitter, we find 129 near-duplicates, 4.93 million suspected spammers and 2.608 billion spam links. Moreover, we cluster the near-duplicates and the corresponding spammers, and analyze the properties of each group.

AKNOWLEDGEMENTS

I would like to present my gratitude to my supervisor Dr. Jianguo Lu for his valuable assistance and support during the past two years.

I also would like to express my appreciation to Dr. Abdulkadir Hussein, Dr. Arunita Jaekel and Dr. Dan Wu. Thank you all for your valuable comments and suggestions to this thesis.

Meanwhile, I would like to thank Hao Wang for collecting the sample of Weibo that is used in this thesis, and Xinyu Liu for the efforts he made on the 2009 Twitter datasets provided by Kwak et al. [17].

Finally, I want to thanks to my parents, my girlfriend and my friends who give me consistent help over the past two years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

Introduction

Online Social Network is a platform to build social networks or social relations among people who share interests, activities or real-life connections. Users in the Online Social Networks (hereafter OSNs) are able to create a public profile, create a list of users with whom to share connections. Among these OSNs users, there exists some spammers. The following screen shots Fig. 1 shows the spam tweets and the spammers in Twitter. They are created to boost other accounts' rank. Spammers send spam contents and links, polluting and distributing the networks. Both users and service providers want to detect and remove the spammers [12, 30, 7].

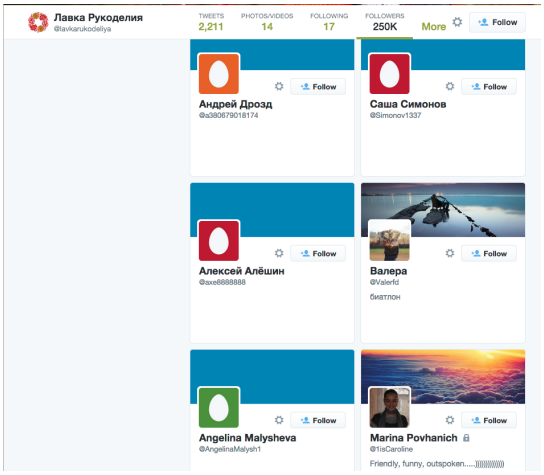


FIGURE 1: Spammers in OSNs

However, it is difficult to distinguish between a spammer and a normal account individually [9]. A spammer account can behave normally with normal screen name, profile picture to avoid being detected and removed by the service providers. These

spammers can be sophisticated robot accounts, or even controlled by human beings directly.

In our work, we propose a novel spammer detection method. We estimate the pair-wise Jaccard similarity among the top-k accounts in OSNs and find the outliners. Then we identify the spammers using these near-duplicates. Meanwhile, we run the agglomerative hierarchical clustering on these near-duplicates and analyze the properties of each cluster. We apply our method on two widely used OSNs: Sina Weibo and Twitter. On Sina Weibo, we find 395 near-duplicates among top 10K users, which lead to 12 million spammers (4.56% of the total users) and 741 million spam links (9.50% of the total links). While on Twitter, we find 129 near-duplicates among top 30K users and 4.93 million spammers(0.42% of the total users) which contribute 2,608 million spam links (3.13% of the total links).

The remainder of this thesis is structured as follows: In chapter II, we review the previous works on spammer detection in OSNs. In chapter III, we address our spammer detection method in detail. In chapter IV, we apply our method on Sina Weibo, which is one of the biggest Online Social Network in China. We address our findings on Sina Weibo and analysis the spammers by groups. In chapter V, we perform star sampling on Twitter, analysis the Twitter networks. Meanwhile, we apply our method on Twitter and identify the spammers in the network. Finally, in chapter VI, we summarize our work and give out the conclusions.

CHAPTER II

Review of The Literature

This chapter reviews the existing works of spammer detection in OSNs. Section 1 reviews two papers which addressed the features of spammers in OSNs. Section 2 reviews three papers that detect spammers in OSNs using machine learning techniques.

1 Analysis the spammers in OSNs

1.1 Suspended accounts in Twitter

Thomas et al. [29] analyze the suspended accounts in Twitter and status the characters of the spammers.

Dataset

The authors collect 1.1 million suspended Twitter accounts in a 7 month period from August 17, 2010 to March 4, 2011. They access Twitter through a privileged account so that more remote calls are allowed. They use *statuses/filter* API to collect a sample of public tweets that contain URLs as well as the account information of the sender. Their collection system receives 12 million tweets per day. In total, 1.8 billion tweets have been collected.

Suspended accounts

To find the suspended accounts, the authors recall the API to check these accounts two weeks after their tweets had been collected. In total, 32.9 million accounts appears

in the collection process and 1.1 million were suspended by Twitter. These accounts send over 80 million tweets that contain 37 million distinct URLs.

Verify

To verify the suspended accounts are spammers, the authors randomly take 100 suspended accounts as a sample then manually verify their tweets and URLs. The authors find that: 93 are suspended for posting scams and unsolicited product advertisement; three are suspended for exclusively retweeting content from major news accounts; the remaining four are suspended for aggressive marketing and duplicate posts. Meanwhile, the authors also take a sample of 200 active accounts and find that 12 are clearly spammers.

Account Analysis

The authors study the properties of suspended accounts:

- Active Duration: 77% accounts are suspended within a day of their first tweet and 92% within three days.
- Tweet Rates: The authors use a sample of 0.1 million accounts to calculate the maximum tweet count for each account and compare it against the account's active duration. The authors point out that the spammers use three different spamming strategies to send the spam content.
- Relationships: The authors analyze the relationship of a sample of 0.1 million active and suspended users. They find that 40% of spam accounts have no followers, while 89% of them have fewer than 10 followers. Meanwhile, the authors point out that the ratio of friends against follower among spam accounts is lower that the one in active accounts.
- Dormancy: The authors find that 56% accounts begin tweeting within the same day the account is created, while 12% lay dormant for over one week and 5% for over one month.

Tweets Analysis

The authors also study the tweets that are sent by suspended accounts and find that they have the following features:

- URLs: of the 37.7 million spam URLs, 89.4% are tweeted once and 34.86% are shortened by *bit.ly*, an URL shorten service.
- Domains: 98.56% suspended accounts tweet fewer than 10 domains in their entire lifetime.
- URL and Domain Reputation: 53% of domains appear more frequently in non-spam tweets than spam comparing to 2.8% of URLs. The authors also suggest to build a domain blacklist to filter the spam tweets.
- API Clients: The authors find out that 58.3% spammers use web API, and 12.39% use *twitterfeed* to send spam tweets.

Spam Campaigns

- Types of spam: The authors classify the spam into three types: affiliate programs, ad-based shorteners and account seller and arbiters.
- Spam Campaigns: *Afraid* is the largest campaign which has 14.525 million tweets and 0.124 million spam accounts. The second one is *Clickbank* which has 3.128 million tweets and 0.016 million spam accounts.

Conclusion

This paper presents the behaviours of spammers on Twitter by analyzing the tweets send by suspended users. The authors find that 89% spammers forgo participation in the social graph; 77% of spammers were suspended within one day. They also study five of the largest spam campaigns on Twitter, which is nearly 20% of the spam in their dataset.

1.2 Link Farm in Twitter

Ghosh et al. [11] address the link farm in Twitter.

Dataset

The authors rely on Twitter's official policy of suspending accounts. Their work is based on the dataset from [4], which contains 54.98 million users and 1.96 billion links as well as 1.76 billion tweets. About 8% users in this dataset are private, which means that only three followers could view their tweets and relationships. The authors ignore these users in their analysis.

Identify spammers

When crawling a suspended account in Twitter, the crawler would lead to the webpage http://twitter.com/suspened. Based on this information, the authors re-crawled the profile page of each user in the dataset in February 2011, and find that 379,340 accounts have been suspended in the interval from August 2009 to February 2011. Then the authors examine tweets posted by these accounts and find 41,352 of those have posted at least one shorted blacklisted URLs. Thus, the authors consider these 41,352 user-accounts as spammers.

Farm links

The authors analyze the followers and targets of these spammers and find:

- Over 15 million Twitter accounts(27% of all users) have been targeted by a small fraction of spammers(0.08% of all users).
- 82% of spam-followers have also been targeted by spammers.

Influence of link farm

The authors compute the PageRank of these spammers in Twitter. They observe that 7 spammers rank within the top 10,000 (0.018% of all users), while 304 and 2,131

spammer rank within in the top 100,000 (0.18% of al users) and 1 million (1.8% of all users), respectively. Thus, the authors state that some spammers acquiried high influence ranks through link farming.

Most farmed links come from few users

The authors analyze the spammed targets who follow spammers, and find most of the spammer links comes from a small number (about 100,000) of Twitter users, who tend to follow back spammer who links to them. The authors create a Twitter account without any profile details, and create links to a set of 500 users randomly from the top 100,000 spam-followers. 13% of the spam-followers respond by following back within 3 days. As a result, the authors' Twitter account ranks among the top-9% of all Twitter users according to the PageRank.

Analysis of link farmers

The authors analyze the link farmers and find:

- Users with low in-degree rarely respond back to spammers. Responsiveness increases with number of followers.
- The authors check the top 100,00 link farmers in July 2011 and find 18,826 have been suspended (hence, possible spammers) and 4,768 are reported as *Not Found*. Furthermore, 235 of the top link farmers have been *verified* by Twitter.
- The authors randomly select 100 accounts from the link farmers and analysis these accounts. In total, 86 accounts are considered to be real. Meanwhile, the authors classify these accounts into two categories: 1) users tweeting on topic like Internet marketing, entrepreneurship, money, and social media. 2) business firms whose tweets attempt to promote their websites.
- Link farmers have 1-2 orders of magnitude higher indexer and outriggers than spammers, and their indegree-to-outdegree ratios are higher than those spammers.

- The authors also point out that top link farmers exhibit very different network connectivity than spammers and a majority of them are not spammers.

- 87% and 79% of top link farmers provide bio and URLs to their external webpages comparing to 25% and 14% of the random users in Twitter. The authors suggest that the top link farmers are active users that make more heavy use of Twitter.

- The authors compute three widely used metrics and find that the majority of link farmers appear within the top 5% of the most influential Twitter users.

Combating link farming

The authors propose *Collusionrank*, a Pagerank-like approach, to combat link farming in Twitter: The static score vector d is initialized by setting the entries that correspond to a set of known spammers to a negative score, and the rest to 0, such that all entries of d sum to -1. Then compute the *Collusionrank* using a method similar to a biased Pagerank computation with $\alpha = 0.85$. Algorithm 1 explains the authors' approach.

Acquiring a low score indicates that a user is colluding with spammers, for which this user should be penalized. Combine Collusionrank with other ranking strategy, such as retweetrank [6], klout, or any topic-sensitive Pagerank-like algorithm[32], can filter out the users who gain high ranks by means of link farming, which is similar to strategies to combat Web spam by combining trust and distrust scores of pages in order to filter out the trustworthy pages from rankings[33].

To evaluate Collusionrank, the authors randomly select a subset of 600 out of the 41,352 spammers and compute the collusion rank scores of all users in the Twitter social network. Experiments use three different randomly selected subsets of 600 spammers yielded almost identical results. The experiments show that more than 40% of the 41,352 spammers appear within the top 20% position in Pagerank, 94% of them are demoted to the last 10% position in Collusionrank. These 94% spammers also appear in the last 10% position in Pagerank+Collusionrank. Specifically, out of

Algorithm 1: Collusionrank from Ref. [11, Algorithm 1]

Input: network, G; set of known spammers, S; decay factor for biased Pagerank,

Output: Collusionrank scores, c

initialize score vector d for all nodes n in G;

$$d(n) \leftarrow \begin{cases} \frac{-1}{|S|} & if n \in S \\ 0 & otherwise \end{cases}$$

/*compute Collusionrank scores*/ **while** *c not converged* **do**

  for *all nodes n in G* **do**

$$tmp \leftarrow \sum_{nbr \in followings(n)} \frac{c(nbr)}{|followers(nor)|}$$

$$c(n) \leftarrow \alpha \times tmp + (a - \alpha) \times d(n)$$

  end

end

return c

the 304 spammers who rank within the top 100,000 Pageranks, 284 have been pushed down to very low ranks.

There are 18,869 out of 100,000 social capitalists ranked within the top 100,000 according to Pagerank. 17,493 of them who follow many spammers are demoted heavily by Pagerank+Collusionrank, while other 1,376 are not affected much. In total, about 20% of the top 100,00 users are pushed down to very low ranks in Pagerank+Collusionrank, while the ranks of the reset users are not affected much.

Conclusion

The authors discover the link farm in Twitter by analyzing the suspended accounts. They consider the suspended accounts as spammers and analysis the ones who follow and been followed by them. Meanwhile, the authors propose a Collusionrank to demote the rank of the link farmers in Twitter.

2 Discover spammers using machine learning techniques

The second type of detection methods are mainly based on machine learning techniques [34].

2.1 Find Spammers in trending topic

Benevenuto et al. [4] address the study of the spammers who send spam in trending topic in Twitter. They study the spammers' attributes and use a Support Vector Machine (SVM) classifier [16] to detect such spammers in Twitter.

Dataset

The authors asked Twitter to while-listed 58 servers that allow them to collect the data from Twitter in August 2009. In total, the authors collected 54,981,152 users, 1,963,263,821 social links as well as 1,755,925,520 tweets. Out of all users, nearly 8%

of the accounts were set private, so the authors ignored these users in their analysis. The dataset can be found in [6].

In this paper, the authors focus on the spammers who targeting trending topics in Twitter. To label the spammers among these 55 million, the authors focus on users who post tweets about three trending topics in 2009. 1) the Michael Jackson's death, 2) Susan Boyle's emergence, and 3) the hashtag "#musicmonday". In total, 8,207 users are labeled, including 355 spammers and 7,852 non-spammers. Among these 7,852 non-spammers, only 710 were selected in their collection. Thus, the total size of labeled collection is 1,065 users.

Identifying spammers' attributes

The authors analyze the labeled collection and analyze the attributes that reflect user behaviour in the system as well as characteristics of the content posted by these users. Finally, the authors consider two attributes sets: content attributes and user behaviour attributes.

Content attributes are properties of the text of tweets posted by users. In particular, number of hashtags per number of words on each tweet; number of URLs per words; number of words in each tweet; number of characters of each tweet; number of URLs on each tweet; number of hashtags on each tweet; number of numeric characters that appear on the text; number of users mentioned on each tweet; number of times the tweet has been retweeted.

While user attributes contains: number of followers; number of followers; fraction of followers per followers; number of tweets; age of the user account; number of times the user was mentioned; number of times the user was replied to; number of times the user replied someone; number of followers of the users followers; number of tweets received from followers; existence of spam words on the users screen name; and the minimum, maximum, average, and median of the time between tweets; number of tweets posted per day and per week.

Detecting spammers

The authors use a non-linear Support Vector Machine (SVM) [16] classifier, which is provided with libSVM, to detect spammers in Twitter using the attributes identified before. The classification experiments are performed using a 5-fold cross-validation. In each test, the original sample is partitioned into 5 sub-samples, out of which four are used as training data, and the remaining one is used for testing the classifier. The process is then repeated 5 times, with each of the 5 sub-samples uses exactly once as the test data, thus produces 5 results. The entire 5-fold cross-validation repeated 5 times with different seeds used to shuffle the original dataset. Results reported are averages of the 25 runs. With 95% of confidence, results do not differ from the average in more than 5%.

The results show that about 70% of spammers and 96% of non-spammers are correctly classified. 30% of spammers are misclassified as non-spammers because their dual behaviours: sharing a reasonable number of non-spam tweets, thus presenting themselves as non-spammers most of the time, but occasionally some tweets that were considered as spam. The authors also point out increasing the parameter J [22] in SVM leads to a higher percentage of correctly classified spammers, but at the cost of a larger fraction of misclassified legitimate users.

The authors also compute the importance of attributes of the spammers. They find out that most important attributes are the fraction of tweets with URLs and average number of URLs per tweet.

Furthermore, authors investigate an approach to detect spam content instead of the spammers using the following attributes: number of words from a list of spam words; number of hashtags per tweet; number of URLs per tweet; number of words; number of numeric characters on the text; number of characters that are numbers; number of URLs; number of hashtags; number of mentions; number of times the tweet has been retweeted; whether the tweet was posted as a reply.

The authors train the classifier and find that about 78.5% of spam and 92.5% of the non-spam tweets are correctly classified. They point out that when users post

non-spam tweets that contain suspect content, the classifier can make mistakes.

Conclusion

The authors crawled the Twitter and obtain 55 million user profiles, all their tweets and social links. They select tweets based on three trending topics and label the collection as spammers and non-spammers. Moreover, the authors study the spammers' attributes and train a SVM classifier to identify the spammers and non-spammers, as well as spam tweets and non-spam tweets.

2.2 Identity spammers using honey-pots

Stringhini et al. [28] address an approach to find the spammers among Online Social Networks using honey-pots and machine learning techniques.

Dataset

On Facebook, the authors first collect some common profile data to build the honey-profiles. They join 16 geographic networks using a small number of manually-created accounts. For each network, they crawl 2,000 accounts at random and randomly mixed these informations (names, surnames, and ages) and create the honey-profiles, while gender is determined by the first name. Each profile is assigned to a network manually. In total, they create 300 accounts on Facebook platform.

Similar to Facebook, the authors get 4,000 accounts and create 300 accounts in total.

On Twitter, the only information required for registration in a full name and a profile name. Thus, the authors simply use the first names and surnames from the other social networks, while the profile name has been chosen as a concatenation of the first and last name plus a random number to avoid conflicts. In total, 300 accounts were created.

These accounts act passively in the networks, which means they do not send any friend requests. But accept all those they received. The authors run scripts

that periodically connect to those accounts and check for activity. These scripts run continuously for 12 months for Facebook (from June 6, 2009 to June 6 2010), and for 11 months for MySpace and Twitter (from June 24, 2009 to June 6, 2010). In total, 4,250 friend requests and 85,569 messages have been collected.

Analysis of collected data

Among the original 3,831 accounts in Facebook, 173 are spammers. On MySpace, there are 8 spammers, while Twitter has 361 spammers in total.

The authors distinguish these spammers into four categories:

- Displayer: Do not post spam messages, but only display some spam content on their own profile pages.
- Bragger: Spammers post messages to their own feed.
- Poster: Send a direct message to each victim.
- Whisperer: Send private messages to their victims. This type of spammers is fairly common on Twitter but none on Facebook and MySpace.

The following table shows the spammers' status in these three networks.

Network	spam per day	avg lifetime	avg friends
Facebook	11	4 days	21
Twitter	34	31	350
MySpace	-	-	31

TABLE 1: Spammers status in Facebook, Twitter and MySpace

The authors summarize two kinds of bot behaviour: stealthy and greedy. Greedy spammers include a spam content in every message they send, while stealthy ones send messages that look legitimate and only once in a while inject a malicious message. Among the 534 spammers that have been detected, 416 are greedy and 98 are stealthy (10 are "displayers" and 20 are "whisperers").

On Facebook, the spammers do not pick victims randomly: most of their victims are male since they promoted for adult websites. Moreover, the spammers seems

to choose the target by looking up the first name of users. For this reason, these gender-aware bots sometimes target female users who happen to have a male name.

Mobile Interface

Most social networking sites have introduced techniques to prevent automatic account generation and message sending. However, these techniques do not work in their mobile interface in order to improve usability. The authors point out that 80% of spammers on Facebook used the mobile site to send their spam message.

Spam profile detection

The authors use machine learning techniques to classify spammers and legitimate users. In the experiment, they only focus on detecting "bragger" and "poster" spammers.

The authors develop six features to detect whether a given profile belongs to a spammer or not:

- FF ratio(R): On Facebook and MySpace, the number of friend requests sent is not public. On Twitter, the ratio $R = follwing/followers$.
- URL ratio(U): the percentage of URLS in the logged messages. $U = messages\_containing\_urls/total\_messages$. While on Facebook, the authors only count URLs pointing to a third party site.
- Message Similarity (S): the similarity $S = \frac{\sum_{p \in P} c(p)}{l_a l_p}$, where P is the set of possible message to message combinations between any two message logged for a certain account, p is a single pair, $c(p)$ is a function to calculate the number of words that shared by two messages, l_a is the average length of message posted by that user, and l_P is the number of message combinations. Thus, the profile sending similar messages will have a low value of S.
- Friend Choice (F): F is defined to detect whether a profile like to used a list of names to pick its friends or not. $F = \frac{T_n}{D_n}$, where T_n is the total number of

names among the profiles friends, and D_n is the number of distinct first names. Legitimate profiles have values of F close to 1, while spammers might reach values of 2 or more.

- Message Sent (M): The authors point out their observation suggests most spam bots send less than 20 messages.

- Friend Number(Fn): The profiles that with thousands of friends are less likely to be spammers than the ones with a few.

Then the authors build two systems to detect spammers on Facebook and Twitter using Weka framework with a Random Forest algorithm [5] as the classifier. They chose this algorithm because it was the one that gave the best accuracy and lowest false positive ratio when performed the cross-validation of the training set.

Spam Detection on Facebook

The geographic network the authors joined in was available at the beginning of authors' study, but discontinued in October 2009. Therefore, the dataset is crawled between April 28th and July 8th 2009. Meanwhile, R feature is not available on Facebook.

The authors train their classifier using 1,000 profiles (173 spammers and 827 manually checked legitimate users). A 10-fold cross validation on this training data set gives out false positive ratio of 2% and false negative ratio of 1%.

Then the authors apply their classifier to 790,951 profiles that collected from Los Angeles and New York networks. They detect 130 spammers in this dataset. Among these, 7 are false positive. The authors claim that the reason for this low number of detected spammers may be that spam bots typically do not join geographic networks. Then the authors randomly pick 100 profiles that classified as legitimate and checked them manually. None of them is spammer.

Spam Detection on Twitter

To train the classifier, the authors pick 500 spam profiles. Some of them are selected from the public timeline to increase diversity. Among the profiles from the public timeline, the authors choose the ones that stood out from the average for at least one of the R, U, and S features. Meanwhile, 500 legitimate profiles have been selected from the public manually as well. The R feature is modified to reflect the number of followers a profile has. The authors also point out that F feature is not useful on Twitter.

The authors perform 10-fold cross validation for the classifier with the updated feature set. The false positive ratio is 2.5% and false negative ratio is 3%.

The authors decide to use their classifier to detect spammers in real time on Twitter. To avoid wasting the limited API calls, they execute Google searches for the most common words in tweets sent by the already detected spammers, and crawl those profiles. Meanwhile, they create a system to allow Twitter users label the spammers. From March 6th, 2010 to June 6th, 2010, they collect 135,834 profiles, 15,932 of those have been determined as spammers. Among these spammers, 75 are reported to be false positives. All the other submitted profiles are deleted. Then the authors randomly pick 100 legitimate profiles and had them manually checked. 6 are false negatives.

Identification of spam Campaigns

The authors visit all the URLs that spammer posted and cluster all the profiles that advertised the same page. The authors list the top 8 campaigns detected on Twitter. Among these campaigns, 3 are observed on Facebook as well.

The authors point out that greedy spammers that send spam with each message are easier to be detected, while a low-traffic spam campaign is not. The spammers from Campaign 1 sent 0.79 messages per day, while the spammers from the second campaign sent 0.08 messages per day on average. The result was that the spammers from Campaign 1 have an average lifetime of 25 days, while the spammers of

Campaign 2 lasted 135 days on average.

Conclusions

The authors create a population of 900 honey-profiles on three major social networks and observe the traffic they received. Then they study the spammers features and use these features to train a classifier. Next, the authors use the classifier to identify the spammers on Facebook and Twitter, respectively. Finally, the authors cluster the spam URLs to find the spam campaigns in both Facebook and Twitter.

2.3 Content based spammer detection method

Wang [30] addresses the work use both content-based and relationship-based feature to identify spammers on Twitter.

Dataset

The author uses Twitter API to crawler the dataset. Non-protected users are selected through *public_timeline* API method. At the same time, the author calls API method *friends* and *followers* to collect detailed information about user's friends and followers. Meanwhile, the author develops a web crawler to collect the 20 most recent tweets of the users that had been selected. In total 25,847 users, count 500K tweets, and around 49M follower/friend relationships are collected.

Features

The author proposes both graph-based features and content-based features to identify spammers on Twitter.

For the graph feature, *reputation* R, is defined as the ratio between the number of friends and the number of followers:

$$R(v_i) = \frac{d_I(v_i)}{d_I(v_i) + d_O(v_i)} \tag{1}$$

The author also proposes four content-based features by looking one's 20 most recent tweets: Number of Duplicate Twitters, Number of HTTP Links, Number of Replies/Mentions and Number Trending Topics.

Analysis

The author trains a Bayesian classifier to identify spammers on Twitter. First, the author labels 500 Twitter user accounts into spammer and non-spammer. Additionally, the author searches "@spammer" on Twitter to collect additional spam data to make the data set contain around 3% spammers.

By analyzing the spammers, the author points out the reputation R of most legitimate users is between 30% to 90%, while most spammers either have a 100% reputation or a very low reputation. The reason that some spam accounts have a 100% reputation is they do not have any friends.

Meanwhile, the author shows that most spammers have multiple duplicate tweets. However, few spammers do not have this feature. Meanwhile, spammers have a higher Mention/Replay counts as well as Link counts than normal users. On the contrary, the number of hashtag of spammers is lower than the one that normal users have. The author points out that the reason trending attack is not common or not even exist due to the spam policy of Twitter.

Evaluation

The author uses 10-fold cross validation to trains the classifier and apply it on 25,817 users that collected before. By checking the sample of the results, the author says that the precision of the spam detection system is $89\% = 348/392$.

Conclusion

In this paper, the author studies the spam behaviours in Twitter. The author proposes the graph-based and content-based features and evaluate them. In the experiment, the author uses these features and a collected dataset to train a Bayesian classifier. The precision of the classifiers is 89%.

3 Summary

In this chapter, we review 5 works that relate to the spammers in OSNs. The previous work by Thomas et al. [29] summarizes the features of the accounts that had been suspended by Twitter, which are possibly spammers. Ghosh et al. [11] find the link farm in Twitter by analyzing the relationship of suspended accounts in Twitter. These two works give us a comprehensive overview of the features as well as the structures of spammers, which lead us a better understanding of the spammers.

Meanwhile, numerous works [30, 28, 4, 18, 34, 27, 9, 20, 37, 8, 7] use machine learning techniques to identify spammers in OSNs. Different classifiers had been used, e.g. Support Vector Machine (SVM) ([4],[35],[22],[18],[20],[7]), Naive Bayesian ([30],[34],[35],[14],[20],[8]) et al. . Moreover, different training dataset had been used: Honeypots ([18] ,[28], [37].), manually labeled ([30], [9],[20],[4],[7], [37]) or blacklist ([34]).

Previous work has classified spammers with high accuracy, but they do have limitations. Previous work identify the spammers by learning spammers' features, which can be manipulated by spammers. For instance, spammers can mix the spam content with the normal one, hibernate for a period before spam the network [29], or hijack the legitimate users to send spam contents or/and links [1].

CHAPTER III

Methodology

In this chapter, we address our spam detection method in detail. In Section 1, we show the Jaccard similarity distribution in random graph and real-world networks. We define the near-duplicates accounts in Online Social Networks in section 2,. In Section 3, we address our method to discover near-duplicates by star sampling. Section 4 describes how we find the suspected spammers in the network by looking the near-duplicates. While in Section 4.2, we define the spammed targets in the Online Social Networks.

1 Jaccard similarity distribution

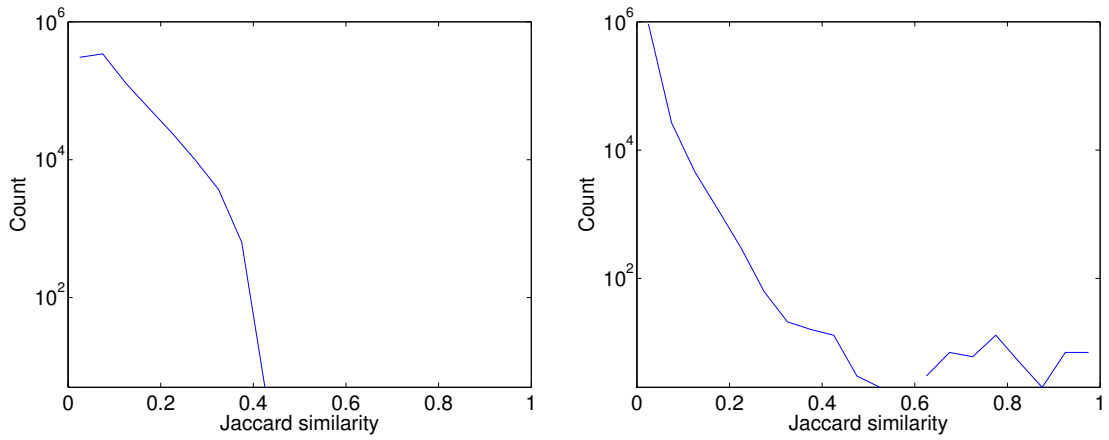
Jaccard similarity, also knows the Jaccard index, is an index for comparing the similarity of two sets. The Jaccard similarity is defined as the sized of the intersection divided by the size of the union of the sets:

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

In a scale free network, the degree follows a power law. Thus, the Jaccard similarity between two nodes would follow the distribution as well. To verify this theory, we generate a random power law network that contains 10,000 nodes with exponent = 1.5 using Snap.py[19], which is a general purpose network analysis and graph mining tool in Python. Then we calculate the Jaccard similarity of the nodes in it. Then, we plot the Jaccard similarity distribution of the random network. Please notice that if we plot the Jaccard similarity of all the nodes in the network, the distribution will have spikes for nodes with small degree. For example, small nodes that have degree of

2 will result in thousands pairs of nodes with Jaccard similarity = 0.5. To avoid these spikes, we only compute the Jaccard similarity among the top users. We compute the pair-wised Jaccard similarity among the nodes that have in-degree larger than 10 and plot the distribution in Fig. 2. As we can see from the figure, the Jaccard similarity of the top nodes in a scale free network follows a power law, which means most Jaccard similarity values are small and only few are large.

When a network has been spammed, in another word, there are large number of spammers that create spam links to boost the rank of their targets, the Jaccard similarity between these targets will be larger than the expected value. Thus, the Jaccard similarity will not follow power-law. In Sina Weibo, a Chinese version of Twitter, there are some users pay fake followers to increase their rank. We choose the ones that have in-degree larger than 0.4 million (1,388 nodes in total), then compute the pair-wised Jaccard similarity of these nodes and plot the distribution with bin-size = 0.05 in Fig. 2. We can see that, when Jaccard similarity is smaller than 0.55, the distribution follows a power-law, which is normal. However, when Jaccard similarity is larger than 0.55, the distribution does not follow power-law anymore. Instead, it has two spikes on Jaccard similarity = 0.75 and 0.9.



(A) Random graph

(B) Sina Weibo

FIGURE 2: Jaccard similarity distribution in random graph and Weibo.

Generally speaking, if two large accounts in OSNs share exactly the same followers in a large number, the followers are most probably artificially created, resulting in

a high value of Jaccard similarity that is times larger than the expected one. For Weibo case, the biggest Jaccard similarity of large accounts should be around 0.55 according to the distribution trend. Instead, there are hundreds pairs of Jaccard similarity larger than 0.55. This is another evidence that the Weibo network has been spammed. Next we will study these "abnormal" accounts to find the spammers in the OSNs.

2 Near-duplicates Accounts

When two books have the same set of n-grams, we say there is a plagiarism; When two web pages receive the same set of hyper-links en masse, there is a web link farm; When two OSNs accounts share thousands or even millions of followers, with uncanny regularity, they are most probably artificially engineered. Borrowing the concept of near-duplicates documents in Information Retrieval, we define the near-duplicates of OSNs accounts as follows:

Definition 1 (near-duplicates) *Two OSNs accounts a and b are called near-duplicates if their Jaccard similarity in terms of their followers is close to one:*

$$J_{a,b} = \frac{|F(a) \cap F(b)|}{|F(a) \cup F(b)|} > \theta, \tag{2}$$

where $F(x)$ is the set of followers of account x, θ is a threshold value that is close to one.

Example 1 (near-duplicates) *In Fig. 3, the black nodes refers to the spammers. These spammers are created to boost the rank of nd_1 and nd_2. As a result, the Jaccard similarity between nd_1 and nd_2 is $J_{nd_1,nd_2} = \frac{|F(nd_1) \cap F(nd_2)|}{|F(nd_1) \cup F(nd_2)|} = \frac{9}{9} = 1$. Which means nd_1 and nd_2 have the exactly same set of followers. Thus, we say accounts nd_1 and nd_2 are near-duplicates.*

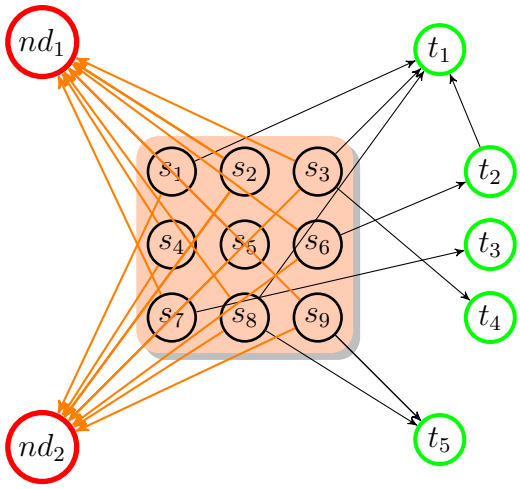


FIGURE 3: An example of spammed network where nd_1 and nd_2 are near-duplicates, $s_1 \sim s_9$ are spammers, $t_1 \sim t_5$ are normal users

3 Discover near-duplicates By Sampling

Discovering near-duplicates among hundreds of million of accounts needs an efficient method. Our new challenge is that most OSNs data is not entirety accessible due to the rate limit of the API. For example, Twitter only allows 15 calls every 15 minutes. It is impossible to get all the users relationships through the API in a short time. Another challenge is to compute the pair-wised Jaccard similarity among the top-k followed accounts. For example, Sina Weibo have over 200 million accounts in 2011. Many accounts have a huge number of followers, in the order of 10^7. Set intersection operation is costly for such large data. Calculating the pair-wised combination among the top-k accounts is out of the question. Although numerous efficient algorithms, such as MinHash [15], are proposed, they are all based on the assumption that the data in its entirety is available. Thereby, we use samples to estimate the Jaccard similarity.

First, we obtain N uniform random sample nodes. Then we extract all the out-links, also known as friends, of these random nodes. These links form a subgraph, where large accounts occur more often in the out-links of the random nodes [31]. This subgraph can estimate the Jaccard similarity of the original graph.

Given two accounts (nodes) n_1 and n_2, suppose that their number of followers

are D_1 and D_2, and their common neighbours are C. In the sampled subgraph, suppose that the sample ratio is $p = n/N$, where n is the number of uniform random nodes in the sample graph, and N is the number of nodes in the original graph. In the subgraph, the expected number of common neighbours is $c = pC$, the expected number of degrees are $d_1 = pD_1$ and $d_2 = pD_2$, respectively. The Jaccard similarity in the original graph is:

$$S = \frac{C}{D_1 + D_2 - C} \tag{3}$$

The similarity in the sample graph is

$$\begin{aligned} s &= \frac{c}{d_1 + d_2 - c} \\ &= \frac{pC}{pD_1 + pD_2 - pC} \\ &= S \end{aligned} \tag{4}$$

Thus, s is the unbiased estimator of S. Next we need to study how large the variance is. c, the common neighbours in the sample graph, follows binomial distribution B(C, p), whose expectation is

$$E(c) = pC. \tag{5}$$

According to the property of the binomial distribution, the variance of c is

$$var(c) = Cp(1-p) \approx Cp. \tag{6}$$

The approximation holds when we assume that the sampling ratio is small. In our experiment, $p \approx 0.005$. The relative standard error (RSE) is

$$RSE(c) = \frac{1}{c}\sqrt{var(c)} \approx \frac{1}{\sqrt{c}} \tag{7}$$

In our experiment, we select a sample probability p that let the minimal c be around 256, $RSE = 1/\sqrt{256} = 0.0625$. Assuming that the distribution of c approximates a normal distribution, we can conclude that the 95% confidence interval for our estimation is within the range of $s \pm 0.125s$. Therefore, the estimation has a high accuracy.

4 Suspected Spammers and Spammed Targets

4.1 Suspected Spammers

Near-duplicates are suspicious in OSNs. Thus, we consider that all the common neighbours of the near-duplicates are suspected spammers. However, not all the suspected spammers are spammers. In fact, two accounts have an expected Jaccard similarity in a network. According to the classic capture recapture model, in a random graph with equal connection probability, the number of expected duplicates D between two nodes i and j is:

$$\begin{aligned} D_{i,j} &= \frac{n_i}{N} \cdot \frac{n_j}{N} \cdot N \\ &= \frac{n_i n_j}{N} \end{aligned} \tag{8}$$

Therefore, the expected Jaccard similarity between nodes i and j is

$$\begin{aligned} E_{i,j} &= \frac{D_{i,j}}{n_i + n_j - D_{i,j}} \\ &= \frac{1}{\frac{n_i}{D_{i,j}} + \frac{n_j}{D_{i,j}} - 1} \\ &= \frac{1}{\frac{N}{n_i} + \frac{N}{n_j} - 1} \end{aligned} \tag{9}$$

In a large network, $n_i \ll N$, $1/N$ is negligible compared with $1/n_i$. Therefore,

$$\begin{aligned} E_{i,j} &\approx \frac{1}{N/n_i + N/n_j} \\ &= \frac{1}{2N} \frac{2}{\frac{1}{n_i} + \frac{1}{n_j}} \\ &= \frac{1}{2N} \langle d \rangle_h \end{aligned} \tag{10}$$

Where $\langle d \rangle_h$ is the harmonic mean of the number of followers of i and j.

However, the OSNs are scale-free networks, the connection probability between two nodes follows power-law. According to our previous work [21], the expected duplicates $D_{i,j}$ in scale-free network is γ^2 larger than the one in uniform random graph, where γ is the coefficient of variation of the node degrees. Thus, the expected Jaccard similarity in OSNs is:

$$E_{i,j} \approx \frac{\gamma^2}{2N}\langle d\rangle_h \tag{11}$$

Example 2 *Suppose in Fig. 3, there are* $N = 1000$ *number of nodes , among them 16 nodes are plotted. And all the remaining nodes not point to nodes* $t_1 \sim t_n$*. The coefficient of variation of the node degrees* $\Gamma = 2$*.*

For nodes nd_1 *and* nd_2*, suppose the expected Jaccard Similarity between* nd_1 *and* nd_2 *is 0.02. Then the expected number of their common neighbours is* $9 \times 9/N \times \Gamma \approx 0.16$*. Therefore, the expected Jaccard similarity is* $0.16/9 \approx 0.02$*.*

Based on this theory, we introduce the $SpammerIndex$ to reflect the probability of a suspected spammer being a spammer.

Definition 2 (SpammerIndex) *Given an account* i*. Let* a *and* b *be the most similar accounts that both have account* i *as their follower. The spammer index of* i *is the deviation from the expected Jaccard similarity, i.e.,*

$$s_i = \begin{cases} J_{a,b} - E_{a,b}, & \exists ab\ s.t. J_{a,b} > \theta; \\ 0, & otherwise. \end{cases} \tag{12}$$

where θ *is the threshold value,* E_{ab} *is the expected Jaccard similarity between accounts* a *and* b*, and* $J_{a,b}$ *is the Jaccard similarity of node* a *and* b*.*

Example 3 *Near duplicates* nd_1 *and* nd_2 *in Fig. 3 share common neighbours* $s_1 \sim s_9$*. Therefore, we consider* $s_1 \sim s_9$ *are suspected spammers, and the SpammerIndex* $s_{1\sim9} = J_{nd_1,nd_2} - E_{nd_1,nd_2} = 1 - 0.02 = 0.98$*.*

4.2 Spammed Targets

In OSNs, each account can have spammers as well as normal accounts as its followers. We define the $SpammedIndex$ to measure the total amount of the spammers received:

Definition 3 (SpammedIndex) *The SpammedIndex* S_j *of an account* j *is the sum of the* $SpammerIndex$ *it receives from its followers. I.e.,*

$$S_j = \sum_{i \in F(j)} s_i. \tag{13}$$

Intuitively, the $SpammedIndex$ measures the total amount of possible spam followers. Popular accounts have large number of followers, and consequently large $SpammedIndex$. To reflect the proportion of the spammed links it receives, we define the $NormalizedSpammedIndex$ as below:

Definition 4 (Normalized SpammedIndex) *The Normalized SpammedIndex NS_j of an account j is the proportion of the spammers links it receives as followers. I.e.,*

$$NS_j = \frac{S_j}{|F(j)|} \tag{14}$$

Example 4 *Spammers $s_1 \sim s_9$ in Fig. 3 share common neighbours $s_1 \sim s_9$ targets nd_1, nd_2 as well as $t_1 \sim t_t$. According to our definition, the SpammedIndex for nd_1 is $S_{nd_1} = 0.98 \times 9 = 8.82$. While the SpammedIndex for t_1 is $S_{t_1} = 0.98 \times 3 + 0 = 2.94$, and the NormalizedSpammedIndex for t_1 is $NS_{t_1} = 2.94/4 = 0.735$.*

CHAPTER IV

Spammers in Sina Weibo

1 Dataset

Sina Weibo (Hereafter Weibo), a Chinese version of Twitter released by Sina corporation in August 2009, has become the most popular Online Social Network platform in China. Weibo provides API service for developers to access their data. In Weibo, there is an up-limit of out-links, Thus, the number of the out-links is not large, and all of them can be obtained from API.

The dataset we use in this paper is crawled by Hao Wang in December 2011 [31]. It contains 1.18 million uniform random nodes and their friends lists. In total, 38,055,283 links are selected. The estimated population of Weibo is 243 million. Therefore, the sample probability is $p \approx 0.486\%$.

2 Near-duplicates Accounts and Suspect Spammers

In our experiment, we set near-duplicates threshold $\theta = 0.9$. Then we estimate the pair-wised Jaccard similarity among top-10K users of Weibo, who have at least 650,000 followers. In total, we find 1,161 pairs of Jaccard similarity, which contain 395 near-duplicates. The following analysis is based on these 395 near-duplicates.

near-duplicates are suspicious in OSNs. We manually check the near-duplicates by looking back these accounts in Weibo using web browsers. By the time we write this thesis, there are 138 accounts have been suspended. And all of these suspended accounts are the targets in the link farm. While the remaining 257 accounts are

either promotions accounts, or business accounts that operated by business company or origination.

These 395 near-duplicates lead us to 11.90 million of distinct spammers and 741.10 million of spam links, which constitute 4.56% and 9.50% of the total numbers of Weibo accounts and links, respectively.

To verify the spammers, we randomly select 100 spammers and manually check them back in Weibo. 81 accounts turn out to be truly spammers. Moreover, 13 accounts are suspended by Weibo. Among the remaining 19 accounts, 14 accounts are inactive accounts and probably been hijacked to re-post spam or generate spam links. Only 5 accounts are legitimate, this is acceptable because our method only detect suspected spammer and give the probability, which is around 0.9 in our experiment, to become a spammer.

3 Spam Targets

In Weibo, the spammers contribute 3,618,065 out-links, which are 9.51% of the total links. Using the index we defined in Chapter III, we can find out the spammed targets in Weibo. The following Table 2 shows the top 20 'polluted' accounts sorted by SpammedIndex, where near-duplicates accounts are not included.

Surprisingly, among the top-20 spammed accounts, there are 3 accounts that belong to Weibo itself, while the others are Chinese celebrities. Another interesting observation is that targets 1266321801, 1761047370 and 1087770692 have almost the same *SpammedIndex*, which is 0.91 million more specifically, their *NormalizedSpammedIndex* are various. The largest one is 0.23 while the smallest one is only 0.06.

ID	Name	Followers ($\times 10^6$)	S ($\times 10^6$)	NS
1642909335	微博小秘书	17.43	2.11	0.12
1654164742	微博名人	6.46	1.77	0.27
1380274560	易建联	5.87	1.24	0.21
1362607654	黄健翔	7.83	1.15	0.15
1656809190	赵薇	11.70	0.96	0.08
1197161814	李开复	9.87	0.94	0.09
1266321801	姚晨	15.42	0.91	0.06
1761047370	大嘴韩乔生	3.99	0.91	0.23
1087770692	陈坤	7.74	0.91	0.12
1182389073	任志强	5.41	0.89	0.16
1182391231	潘石屹	7.75	0.88	0.11
1682352065	周立波	9.64	0.80	0.08
1658688240	手机微博	3.02	0.80	0.26
1686326292	梁咏琪	5.94	0.79	0.13
1670071920	史玉柱	4.44	0.76	0.17
1222713954	陈志武	3.32	0.75	0.22
1470110647	于嘉	3.67	0.74	0.20
1192515960	李冰冰	8.85	0.69	0.08
1282005885	蔡康永	12.29	0.68	0.06
1650569064	朱骏	3.32	0.68	0.20

TABLE 2: Top 20 'polluted' accounts sorted by SpammedIndex. near-duplicates accounts are not included.

4 Cluster of Near-duplicates Accounts and Spammers

In the experiments, we find an interesting group called Antique shops(文玩天下). The structure of this group is illustrated in Fig. 4. This group has 24 near-duplicates, all of them have the same followers with the amount of 0.2 million. Among these 0.2 million spammers, 96 percent have no followers at all; 81.68 percent have the same out-degree (25, pointing to the near-duplicates only); 86 percent are registered on the same two months (April and May 2011); 80.06 percent are registered from Beijing; zero percent are cell phone users or verified users; 95.57 percent have never sent any messages.

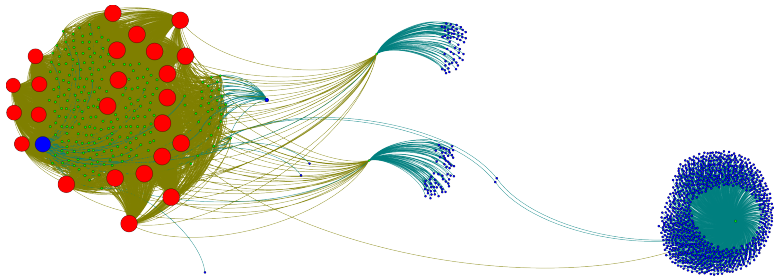


FIGURE 4: The antique shop group: near-duplicates (the red nodes) are followed by the same group of spammers (green nodes). Those spammers are mostly of zero incoming links. Most of the spammers point to the near-duplicates only, with a few exceptions that point to a larger number of other nodes (the blue nodes).

Next we plot the 395 near-duplicates and top-200 spammed targets in Fig. 5. The red nodes are near-duplicates and the black node are spammed Targets. The edges are Jaccard Similarity between two nodes: red one represent the Jaccard similarity that larger than 0.9 while the blue one represent the Jaccard similarity that larger than 0.5.

We can see that the spammers pointing to different groups of targets, which indicates they are created by a variety of spam producers. To find out their origins, we run the agglomerative hierarchical clustering on these near-duplicates using (1-JaccardSimilarity) as the distance function. Fig. 6 shows the resulting dendrogram,

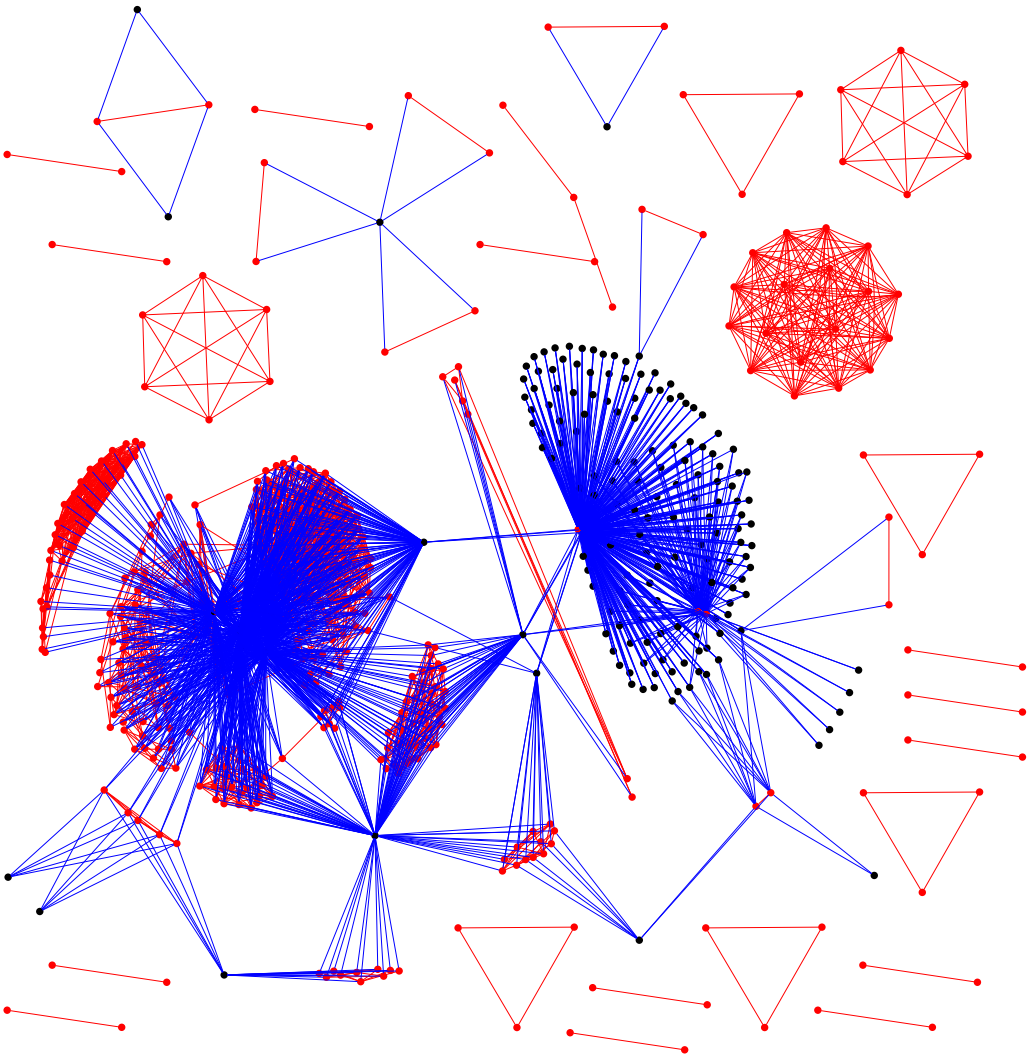


FIGURE 5: 395 near duplicates and top 200 spammed targets.

which is created by the unweighted pair group method with arithmetic averages (UPGMA) linkage.

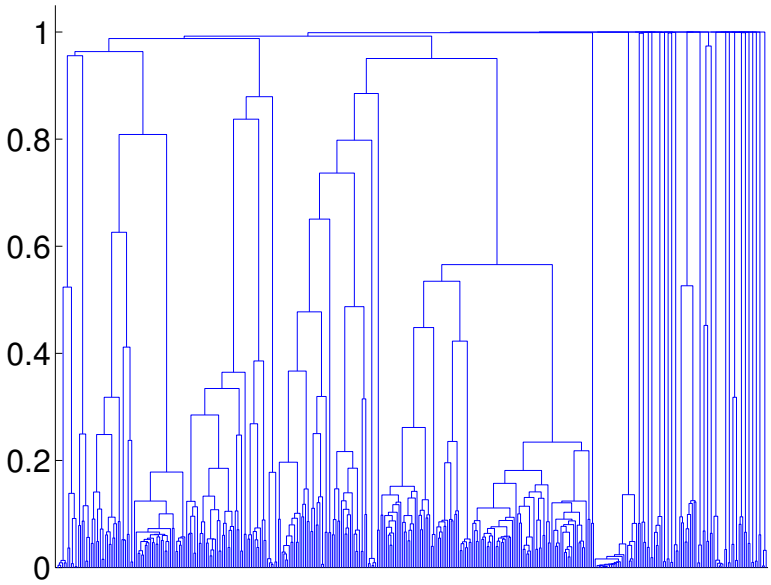


FIGURE 6: Dendrogram of the clustering result of the 395 near-duplicates.

When cutting the dendrogram at the value of 0.85, we find 34 clusters. To verify the result of clustering, we plot the relationship between the near-duplicates and random spammers before and after the clustering in Fig. 7 and Fig. 8 respectively. For the near-duplicates, we observe that:

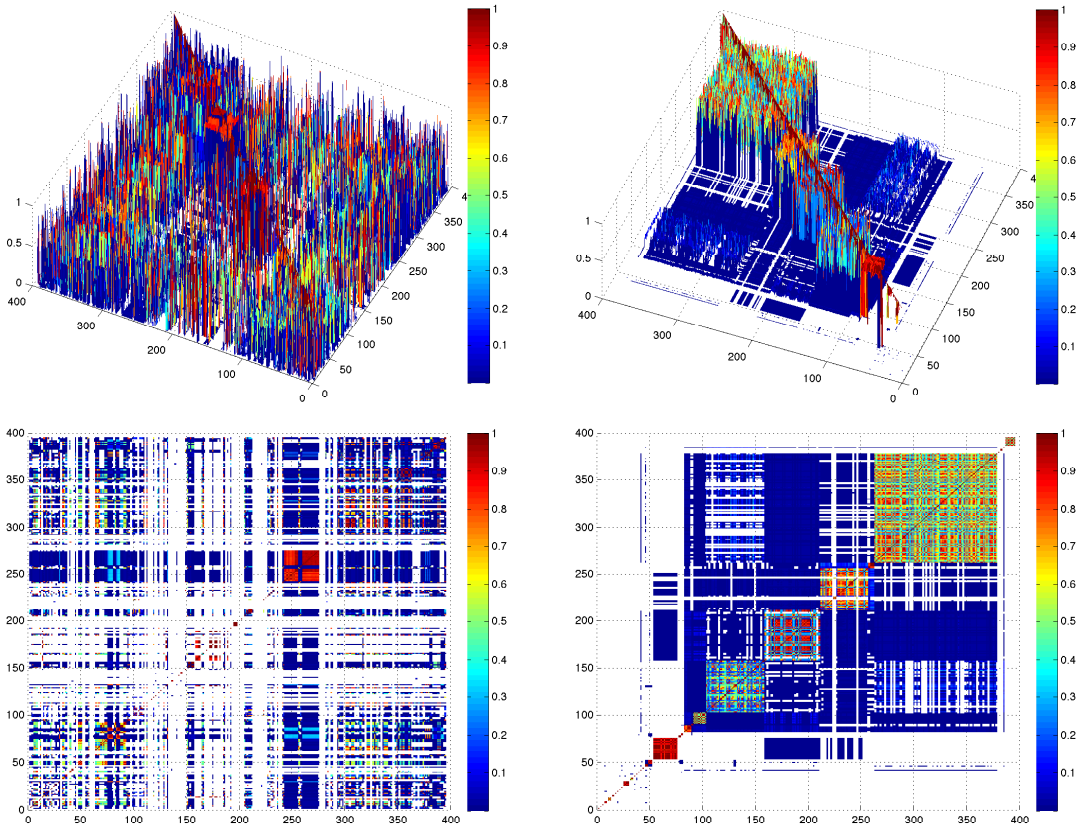
1. some near-duplicates IDs are contiguous (e.g., the red block around IDs 250 in Panel (A) of Fig. 7), indicating that they are created around the same time;

2. When near-duplicates are clustered, there are small groups of near-duplicates that contain only a few members, mostly in the left lower corner in Panel (B) of Fig. 7;

3. there are several large clusters. The largest one (cluster 30) contains 117 near-duplicates, and is depicted in the upper-right corner of Fig. 7;

4. Some clusters (the first and last a few clusters) are completely isolated from the remaining near-duplicates.

This is another evidence that they are not normal accounts. Large accounts normally have some overlapping followers.

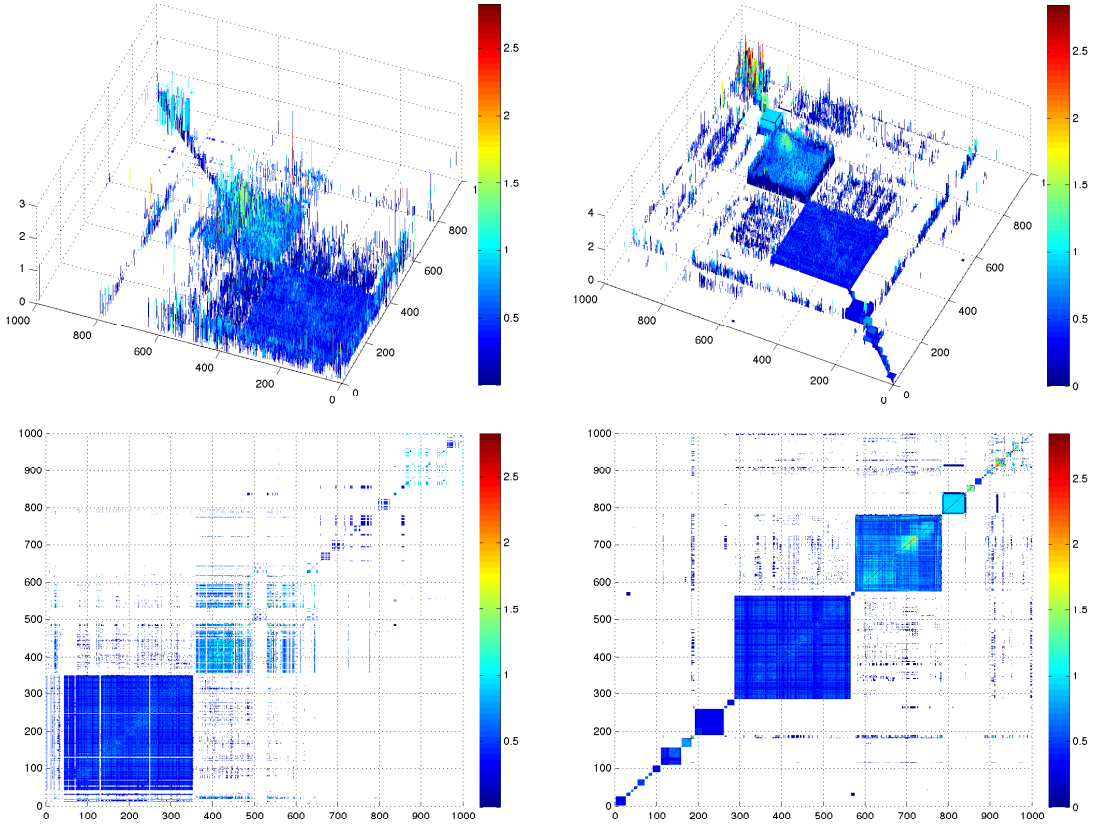


(A) Sorted by IDs (B) Sorted by clusters

FIGURE 7: Jaccard similarities of 395 near-duplicates.

Next, we explore the relationship of the spammers of those clusters depicted in Fig. 8. There are 12 million of spammers, which are too large to visualize. We select 1000 spammers uniformly at random, and plot the size of common followers between the random spammers in Fig. 8 . Panel (A) is sorted by ID, and Panel (B) by clusters. From Panel (A), we can observe that spammers are roughly grouped by their IDs, indicating different spam producers create their spam accounts at different period of time. After clustering, we highlight the following observations: 1) there are two large groups of spammers, corresponding to WeiboAssistant (cluster 16) and DatingGroup (cluster 18). Note that these two clusters contain only five near-duplicates (two for WeiboAssistant and three for DatingGroup). Thus, they are not discernible in Panel

(B) of Fig. 7. Yet they have millions of spammers, thus forming two large blocks in Panel (B) of Fig. 8; 2) a large cluster of near-duplicates (cluster 30) has a few of spammers as shown in the upper corner of panel B. Yet spammers in cluster 30 are highly integrated by sharing hundreds of common followers; 3) Since clusters are sorted in terms of in-degrees, spammers in each cluster become increasingly more integrated; 4) Spammers in most clusters have the similar number of common followers;



(A) Sorted by IDs (B) Sorted by clusters

FIGURE 8: Common friends of 1K random spammers plotted in log10 scale.

5 Structures of Spammers

In the last section, we split the 395 near-duplicates into 34 clusters. The details of the clusters, including the statistics of the spammers and the links to the web page depicting the details of each near-duplicates, are listed in Table 3. It lists the clusters

in the increasing order of the average in-degrees of the spammers. In addition to the in-degrees, we also list the average out-degrees, the number of spammers, the number of spam links, and the number of near-duplicates in each cluster. For each cluster we also provide a web page that describes the details to these near-duplicates.

Next, we summarize their properties in Fig. 5 in contrast to the random accounts in the first column. For random accounts in Weibo, the in- and out-degrees have a heavy tail, just the same as Twitter and many other social networks. The message counts for each account also has a long tail resembling a power law. The fourth row describes the spam links vs. in-degree. For random accounts, it shows that large accounts (accounts with large in-degree) tend to receive more spam links. Row 5 depicts the location distribution, while row 6 shows the percentage of the accounts that are created in each of the 27 months. For random accounts, almost the same amount of new accounts are created during the last 10 months.

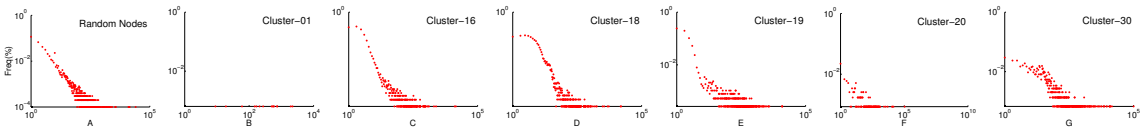
By studying the properties of these 34 clusters, we describe the following four representative types of spammers, ranging from simple complete bipartite graph to complex link farm.

- Complete Bipartite Graph: Spammers and their targets are disjoint. The number of spam targets is very limited, and every spammer connects with every target. For example, Fig. 10 (A) is a random sample of the spammers in cluster 1 (Love Shopping), where every spammer follows only two spam targets. Their in-degree is 0.00, out-degree is 2.00. Most probably the spammers are created for the sole purpose to boost the follower number of these two accounts. Accounts in cluster 1 are obviously spammers as can be shown by the column two in Fig. 5: their in-degrees are mostly 0, out-degrees are two, most accounts never post any messages, and their creation time and place are also the same.

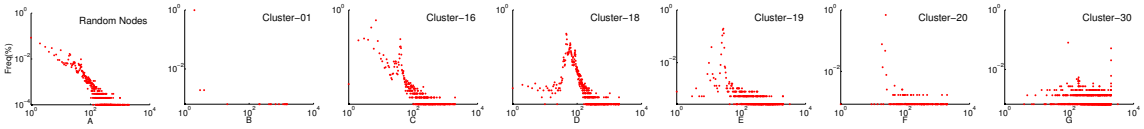
- Bipartite Graph: Spammers and their targets are disjoint. Spammers aim at more spam targets. Fig. 10 (B) illustrates the spammers in Gif Animation. Every spammer follows multiple spam targets, but its out-degree is a constant (4 for these spammers).

Cluster	Spammers				#near duplicates	Name
	In-deg (avg)	Out-deg (avg)	#Spammers ($\times 10^6$)	#Links $\times 10^6$		
1	0.00	2.00	0.31	0.64	2	Love Shopping
2	0.00	4.00	0.06	0.27	2	Android Group
3	0.00	4.01	0.06	0.26	2	Gif Animation
4	0.17	5.73	0.07	0.44	3	Campus Chongqing
5	0.20	5.62	0.10	0.61	4	Campus Shangrao
6	0.43	5.69	0.22	1.30	3	Spam Group
7	0.84	7.29	0.09	0.70	2	Campus Jinan
8	0.95	6.09	0.11	0.71	2	Campus Xian
9	1.01	4.11	0.24	1.01	2	Mobile Neimengu
10	1.33	6.58	0.62	4.09	2	Mobile Winner
11	2.85	8.88	0.27	2.42	6	Liaoning Telecom
12	3.40	56.24	0.10	5.83	4	3G
13	3.58	6.79	0.86	5.84	2	Zhejiang Telecom
14	4.38	22.19	0.09	2.07	2	Mobile Dream
15	5.29	4.66	0.22	1.02	2	Love Hubei
16	7.15	18.96	3.49	66.24	2	Weibo Assistant
17	9.12	7.55	0.13	0.95	2	Mobile Marketing
18	10.22	69.19	2.49	172.71	3	Dating Group
19	10.68	36.43	0.65	23.63	5	Telecom Animation
20	15.51	29.41	0.20	5.96	24	Antique Shop
21	16.91	7.69	0.22	1.68	2	Telecom Jilin
22	23.97	17.53	0.05	0.95	3	Telecom Wuhan
23	41.56	213.27	0.08	17.49	7	Naming
24	44.55	99.14	0.12	12.05	2	Photo
25	47.58	130.10	0.16	20.19	12	Deleted
26	60.30	675.34	0.28	190.22	55	Green Tea etc.
27	63.54	204.11	0.27	42.18	53	Pets etc.
28	67.84	270.69	0.19	52.11	45	Wedding etc.
29	78.02	333.40	0.07	23.24	6	Deleted
30	93.05	924.70	0.19	177.81	117	Sydney Coupon etc.
31	125.48	484.12	0.06	28.38	2	Spam Farm
32	133.09	134.46	0.07	8.68	3	Software
33	203.13	615.68	0.07	42.03	2	Health
34	251.16	504.65	0.08	43.55	10	Chinese medicine etc.
Sum			11.90	741.10	395	
Mean	14.01	61.83				

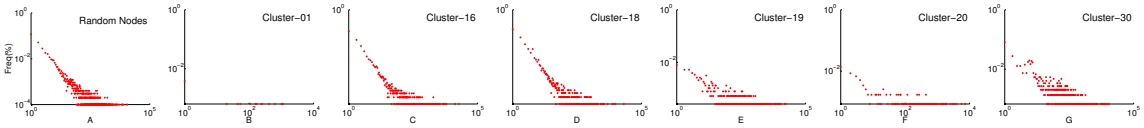
TABLE 3: 34 clusters of near-duplicates and the statistics of their spammers, sorted by the increasing order of their average in-degrees. Each cluster has a clickable URL link that points to our web page showing the details of the near-duplicates and their similar accounts.



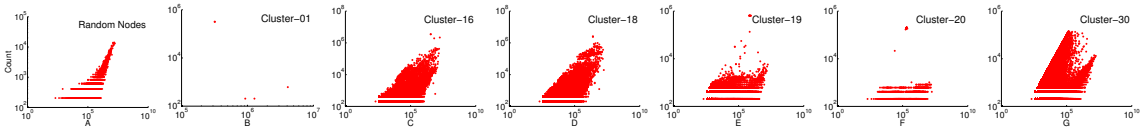
(A) In-degrees



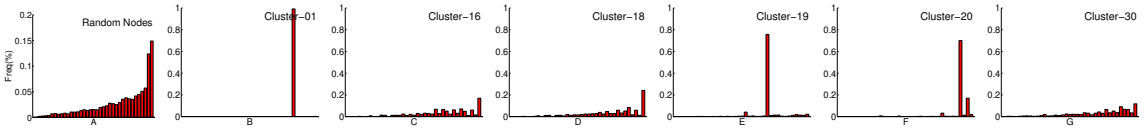
(B) Out-degrees



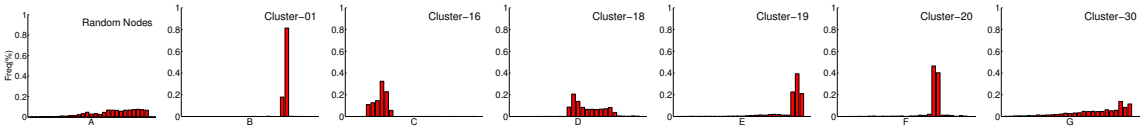
(C) Messages



(D) Target in-degrees



(E) Locations

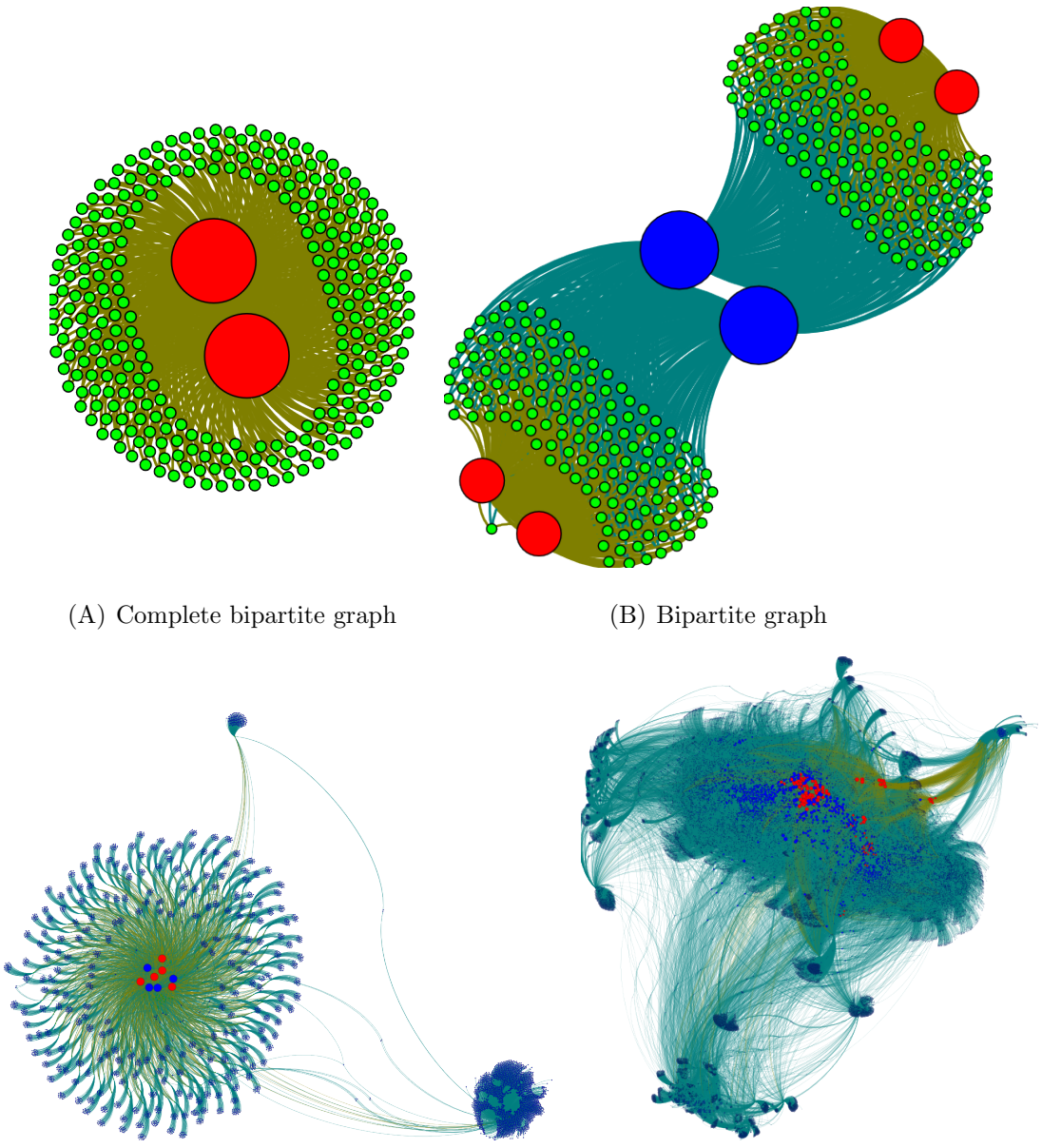


(F) Creation Month

FIGURE 9: Properties of six clusters (columns 2 to 7), and a comparison to that of the random accounts (column one).

- Power law in-degree: This kind of spammers are more sophisticated in that they try to blend in by making their in-degrees following a power law, just like most networks [24]. Spammers follow their main targets as well as some other random accounts, so that it is not obvious to be detected. In contrast to the zero in-degree in the bipartite graphs, these spammers receive follow links. Interestingly, the in-degrees of spammers follow a power law. However, most of their out-degrees are the same, and their frequencies follow a log-normal distribution. Fig. 10 (C) illustrates such an example spammer group (Telecom Animation) where many spammers (23%) have out-degree of 28. Most of their targets are disconnected, while their main targets remain to be an obvious small set.

- Link farms: The main targets are no-longer limited to a few accounts. Spammers and their targets are closely knit – spammers are the targets of other spammers. Fig. 10 (D) illustrates a spammer cluster that involves 117 near duplicates, and many other spammer targets. Spammers typically have the maximal out-degree that is allowed by the system.



(A) Complete bipartite graph

(B) Bipartite graph

(C) Power law in-degree

(D) Link farm

FIGURE 10: Four types of spammers.

CHAPTER V

Spammers in Twitter

Twitter is one of the most popular Online Social Networks in the world [3]. Millions of people share their lives and connections in Twitter. In this Chapter, we perform star sampling on Twitter. Then we estimate the Jaccard similarity among the top-30K users, identify the near-duplicates and the corresponding spammers.

1 Datasets

Twitter is a developer-friendly platform who provides Application Programming Interface (API) that allows us to crawl and collect data. In the past, Twitter provides API Whitelist which allow developers collect data from Twitter without query limitation. However, this feature has been revoked in February 11th 2011 [26]. After that, only few queries can be processed in a window of 15 minutes. To speed up our crawling process, we use multiple develop accounts to collect the data we need.

The API function we used is our crawler are:

- users/lookup:
  - This function can query up to 100 users in one request.
  - Only 60 requests can be processed in 15-min window.
  - A requested user will not be returned if it is unknown, suspended, or deleted.
  - If no users satisfies the condition, a HTTP 404 will be thrown.
- friends/ids:

– This function returns up to 5,000 friends IDs for the queried user.

– Only 15 requests can be processed in 15-min window.

1.1 User ID Space

First, we need to identify user ID space of Twitter. Then we can obtain the uniform random accounts by generating a random number within the ID space. This method is used to obtain uniform random nodes (accounts) from Facebook [13], Youtube [36] and Weibo [10, 31]. Assume that Twitter use the user ID as the primary key of the user table, which is generally auto increasing to avoid duplicates. We randomly select some users in Twitter, and plot their user IDs against their creation time in Fig. 11. The X-axis is the user IDs we get from twitter, and the Y-axis is the creation time formatted in Unix time. From the plot we can see that user ID is positively related to the creation time. Thus, we say we can determine the ID space by obtaining the maximum user ID in Twitter.

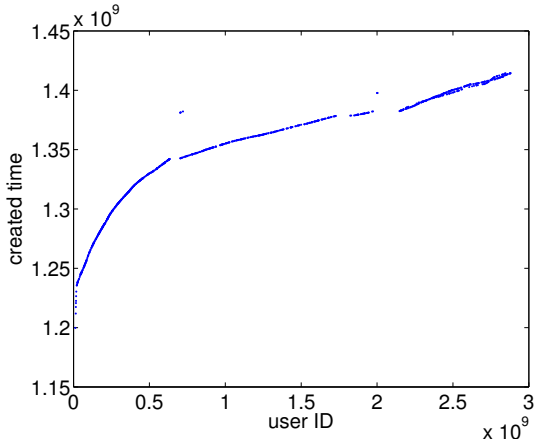


FIGURE 11: User ID against Creation Time.

Twitter API provides a */user/lookup* function that allows us to query up to 100 users in one single query. Using this feature, we randomly generate 100 user IDs then send them to the API. The Twitter server will respond the user information. If none of them associates with a user, then API will return HTTP 404 error. We log the maximum user ID among the ones in the last process, and randomly generate

100 user IDs that larger than the logged maximum ID. Repeat this progress until the maximum user ID we found is created within 10 minutes. The process can be explained by Algorithm 2.

Algorithm 2: Find maximum ID

Input: a legitimate Twitter user u, maximum expected user ID r

Output: maximum user m

let $m = u$;

while $m[created\_at] > currentTime - 10mins$ **do**

  let q be an array that contains 100 random integers in the range of $(m[id], r)$;

  $validUsers \leftarrow API/user/lookup(q)$;

  let m be the user in $validUsers$ that has a maximum user ID;

end

return m;

We launch our crawler on Nov 10th, 2014 and find that the maximum user ID of that time is 2895209485. Our experiment is based on the ID space of 0 to 2895209485.

1.2 Uniform random nodes and star sampling

To obtain an uniform random node, we can randomly select an ID from the ID space. However, not all every ID is associated with a user. We need to test it in Twitter. In our research, we generate 100 random integers within the range of 0 to 2895209485 then test these integers using through the API, then record the valid ones as uniform random nodes. Repeat this progress until the amount of the uniform random nodes reaches our needs.

First, we sent 249,23 queries which contain 2,492,300 random user ID ranging from 0 to 2895209485. Among these 2,492,300 user IDs, 1,000,001 were associated with users. Next we calculate the probability of an integer ranging from 0 to 2895209485 associated with a user, which is $1,000,001/2,492,300 \approx 40.12\%$. Thus, the population of Twitter is $2,895,209,485 \times 40.12\% \approx 1,161,662,873$. Thus, the ratio of

uniform random sample is $p = 1,000,001/1,161,662,873 \approx 0.086\%$.

Next, we use these 1 million uniform random nodes as seeds and perform star sampling by fetching all the out-links of them. Among these seeds, 53,074 are set as "private", which means only there friends can see their relationship and tweets. In our research, we ignore these users as other researchers did [11, 4]. In total, we obtained 66,036,468 relationship.

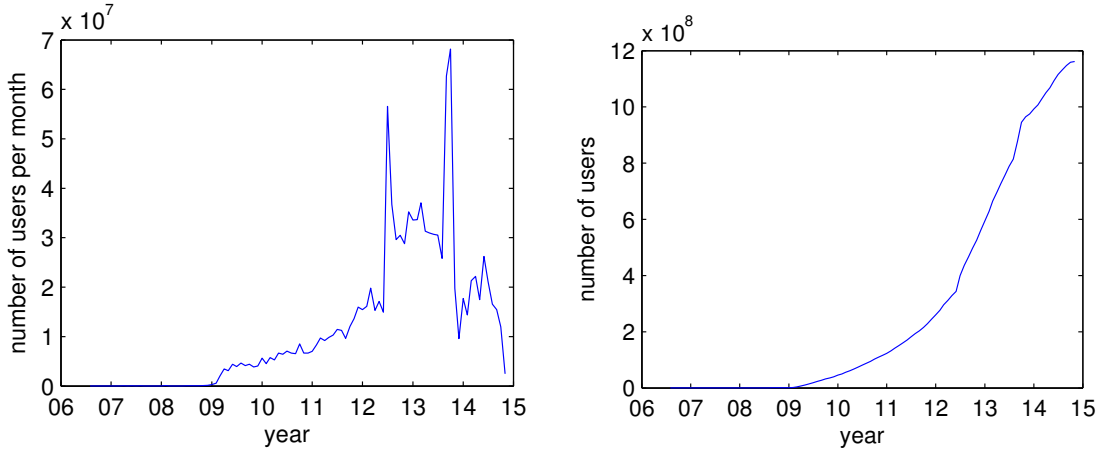
2 Analysis Twitter Datasets

Before going into the spam detection section, we would like to take a overview of the Twitter Datasets. Back to 2009, Kwak et al. [17] crawled the entire Twitter network that contains 41.7 million users and 1.47 billion relationship. Meanwhile, Myers et al. [23] address the structure of the twitter follow graph based on a snapshot of Twitter of 2012, which contains 175 million active users and approximately 12 billion edges. With the growing of Twitter website, Twitter has 284 million monthly active users, 500 million Tweets per day in 2014 [2]. Although we can not access the entire Twitter network, we still can estimate some properties[31] of Twitter using the sample we obtained. This will not only provide us a comprehensive overview of the growing of Twitter, but also give us a better understand of the current Twitter networks.

2.1 Creation time

When we determine the user ID space, we notice that Twitter grows extremely fast: the maximum user ID we find in November 10th is 2895209485, and when we re-launch the program on January 7th, the maximum user ID we find is 2966745190. To find how fast the population of Twitter grows, we estimate the amount of register users of Twitter in each month 8 years (June 2006 to October 2014) and plot the results in Figure 12. In the first 30 months (before February 2009), the number of register users in Twitter grows very slow. After that, Twitter attracts more and more users and the number reaches the first peek on the 71st month since it released, which

is July 2012. The second peek appears on the 86th month (October 2013). However, the number of new users per month continually falls since the June 2014, the 94th month since it founded, and reaches the lowest point at October 2014.



(A) Monthly registered users (B) Amount of registered users

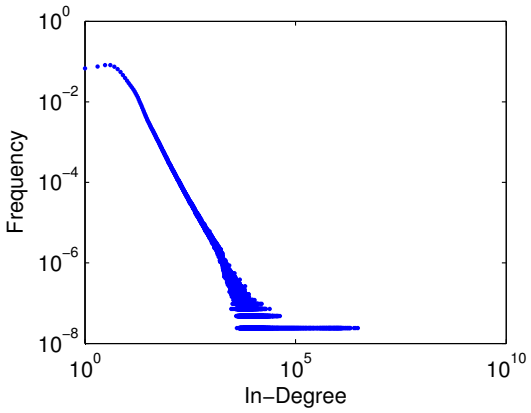
FIGURE 12: User Creation Time in Twitter.

2.2 Degree Distribution

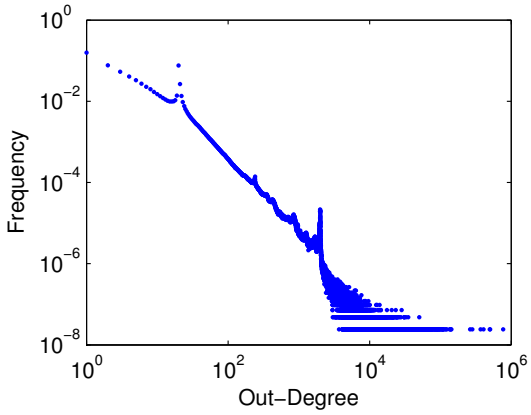
Since the relationship of a user on Twitter is directed, the user's information contains both in-degree (the number of users who follow them) and out-degree (the number of users who they follow). In our research, we use the dataset from the work by Kwak et al. [17] to make the comparison. The degree distribution of Twitter in both 2009 and 2014 can be found in Fig. 13 a) and b), respectively. Meanwhile, we also study the degree distribution of 'private' user of 2014 (Fig. 13 a) and c))

The Figure shows that the distributions of in-degree and out-degree of Twitter in both 2009 and 2014 follow power-law just as we expected. Both dataset have spikes on out-degree of 2000. This is because Twitter has a restriction that a user can not have more than 2,000 friends unless it has more than 2,200 followers[23].

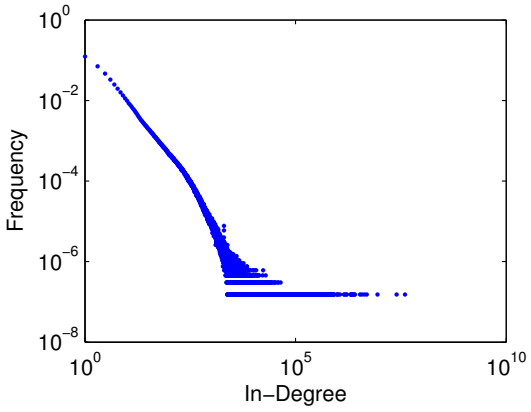
We show the mean and median in/out -degree in the following Table 4. The average in-degree in 2009 is 36.61 and the one in 2014 is 112.50, which is three times larger than the previous one. While the out-degrees in 2009 and 2014 are 41.14 and



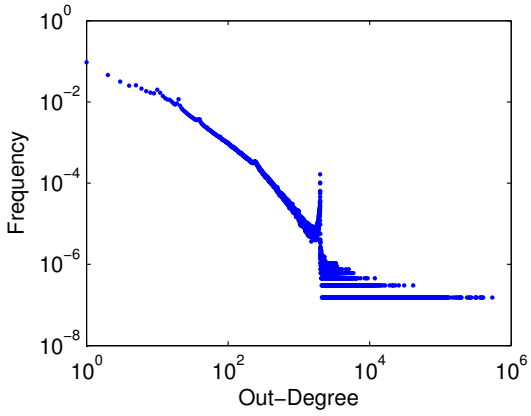
(A) In-degree in 2009



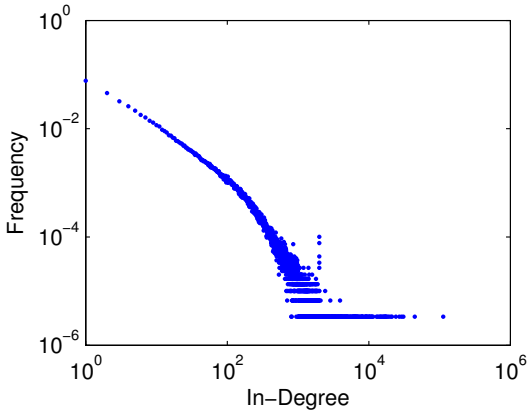
(B) Out-degree in 2009



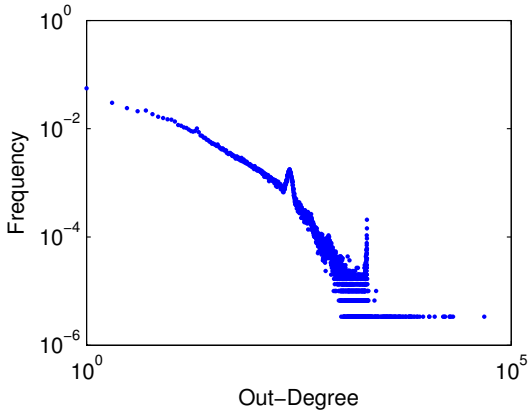
(C) In-degree in 2014



(D) Out-degree in 2014



(E) In-degree of private users in 2014



(F) Out-degree of private users in 2014

FIGURE 13: Twitter degree distribution in 2009 and 2014.

72.04, respectively. As to the private user, the average in-degree is almost twice lower than the global value, 64.09 against 112.50. However, they have a higher out-degree, which is 113.09 comparing to 72.04.

Data	in-degree		out-degree	
	mean	median	mean	median
Twitter 2009	36.61	7	41.14	8
Twitter 2014 (global)	112.50	1	72.04	7
Twitter 2014 (private)	64.09	5	113.09	32

TABLE 4: Mean and Median degree of Twitter in 2009 and 2014

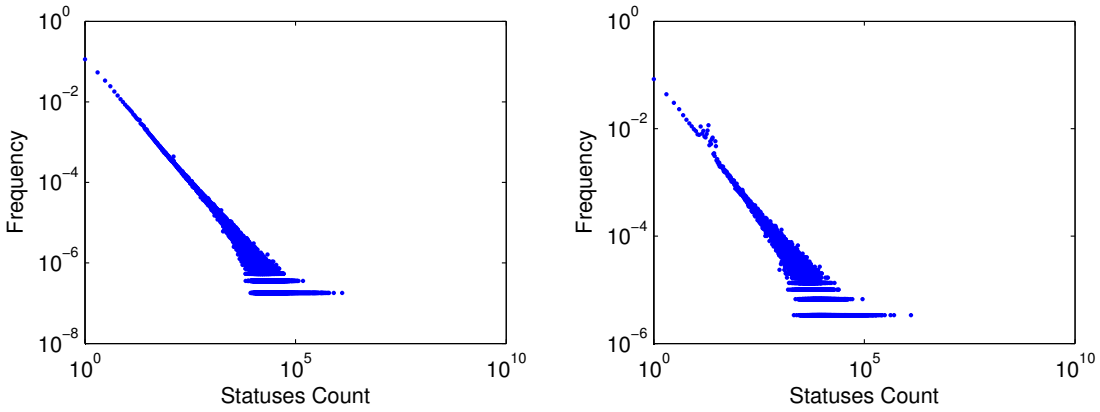
Next, we estimate the top followed users of Twitter in 2014 showed in Table 5. Back in Summer 2012, Lady Gaga was the most followed user on Twitter[23]. While in 2014, she falls to the rank of 6 and Katy Perry becomes the most followed user, who has 62.8 million followers on Twitter. Barack Obama had 21 million followers and followed more than 600K people in 2012 while now the number grows to 52.6 million and 646K, respectively.

Rank	User	in-degree	out-degree	#tweets
1	Katy Perry	62.8M	161	6.3K
2	Justin Bieber	58.8M	172K	28K
3	Barack Obama	52.5M	646K	12.9K
4	Taylor Swift	50M	156	3.1K
5	YouTube	47.8M	851	12.3K
6	Lady Gaga	43.4M	133K	6.3K

TABLE 5: Top followed users in Twitter.

2.3 Statuses

According to Twitter website, 500 million Tweets are sent per day[2]. The average statuses count of Twitter is 429.90 and the median is 1. The average and median statuses count for the private users are 1122.34 and 19, respectively. The distribution of statuses follows power-law, which can be found in Fig. 14.

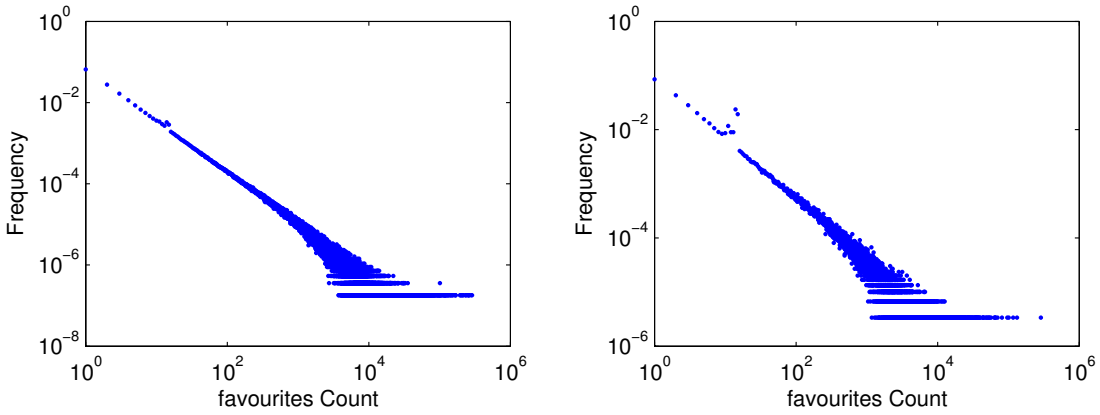


(A) All users (B) private users

FIGURE 14: Statuses counts of all users and private users on Twitter

2.4 Favourites

In Twitter, user can add their favourites tweets in their favourites list. The average statuses count of Twitter is 69.54 and the median is 0. While the number for 'private users' are 168.28 and 1. The distribution of this feature is plotted in Fig. 15. The x-axis is the number of favourites while the y-axis is the corresponding count percentage. The plot is set as log-scale. The left one is the distribution of all users in Twitter while the right one is for private users.



(A) All users (B) private users

FIGURE 15: Favourites counts of all users and private users on Twitter

2.5 Conclusions

Twitter is the most popular Online Social Networks in the world. Due to the new policy of Twitter API, we can not crawl the entire Twitter network like Kwak et al. [17] did in 2009, or get the active users from Twitter like Myers et al. [23] did. Instead, we determine Twitter user ID space and perform uniform random sample and star sample on Twitter. Our estimation reports that the population of Twitter is 1.16 billion, which is 4 times larger than the 284 million monthly active user report by Twitter Inc[2]. Meanwhile, we found that 5% users in Twitter have been set to private. The private users have larger out-degree than the normal users. Meanwhile, they send have more tweets and favourites tweets in Twitter. This observation indicates that these private users are 'heavy users' of Twitter.

3 Near-duplicates Accounts and Suspect Spammers

In our experiment, we set near-duplicates threshold $\theta = 0.9$. Then we estimate the pair-wised Jaccard similarity among top-30K users of Twitter. In total, we find 397 pairs of Jaccard similarity, which contain 129 near-duplicates in total.

When visiting these near-duplicates by web browser, we find that 107 of them are business accounts who post news or promotion tweets. Most of the tweets send by these accounts contain URLs. 3 accounts have stopped posting for over 6 months. Moreover, there is one account acts like robot who continually sends message with a certain pattern. Meanwhile, there are 16 accounts have no tweet, or very few (less than 100 in two years) in their entirely lifetime and yet, they have followers in number of 10^5. Only 2 accounts seem to be owned by human, however, all their tweets are the messages replied to other users. Thus, we say the near-duplicates we find in Twitter are suspicious so that we call the ones that follower them are suspected spammers.

In total there are 4.93 million of distinct spammers and 28.335 million of spam links on Twitter, which constitute 0.424% and 0.0408% of the total numbers of accounts and links, respectively.

4 Cluster of Near-duplicates Accounts and Spammers

We apply the same cluster method explained in Chapter IV on these 129 near-duplicates. The following Fig. 16 shows the resulting dendrogram.

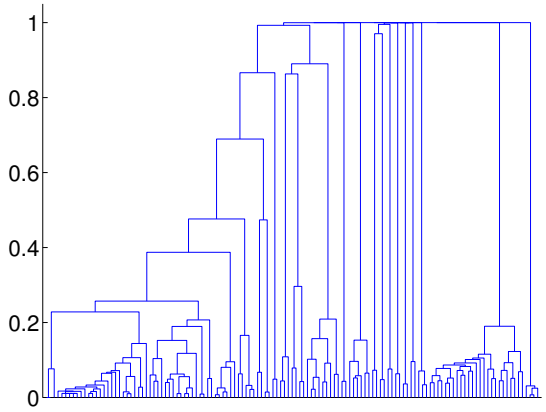


FIGURE 16: Dendrogram of the clustering result of the 129 near-duplicates.

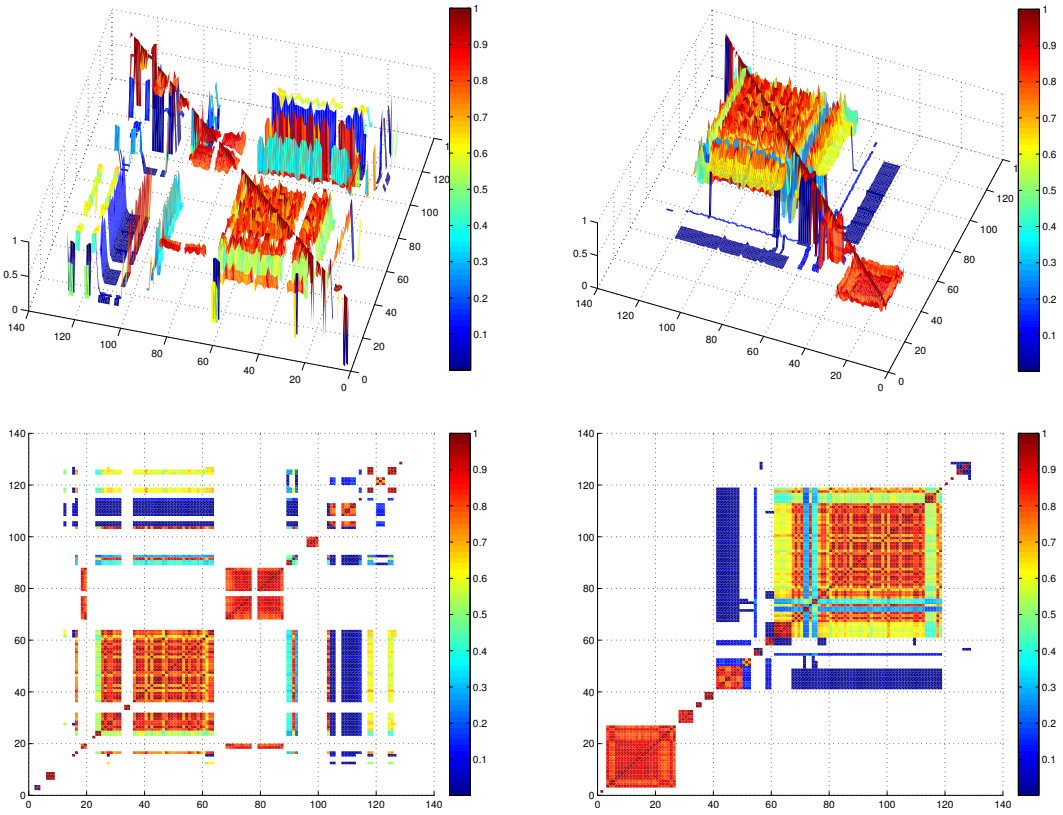
When cutting the dendrogram at the value of 0.85, we find 16 clusters. To verify the cluster, we plot the Jaccard similarity of 129 near-duplicates sorted by both ID and clusters in Fig. 17. Meanwhile, we randomly select 1000 spammers and show the common followers in Fig. 18

The details of the clusters, including the statistics of the spammers and the links to the web page depicting the details of each near-duplicates, are listed in Table 6. It lists the clusters in the increasing order of the average in-degrees of the spammers.

The following Table 4 lists the most appeared attributes of each cluster.

5 Structure of Spammers

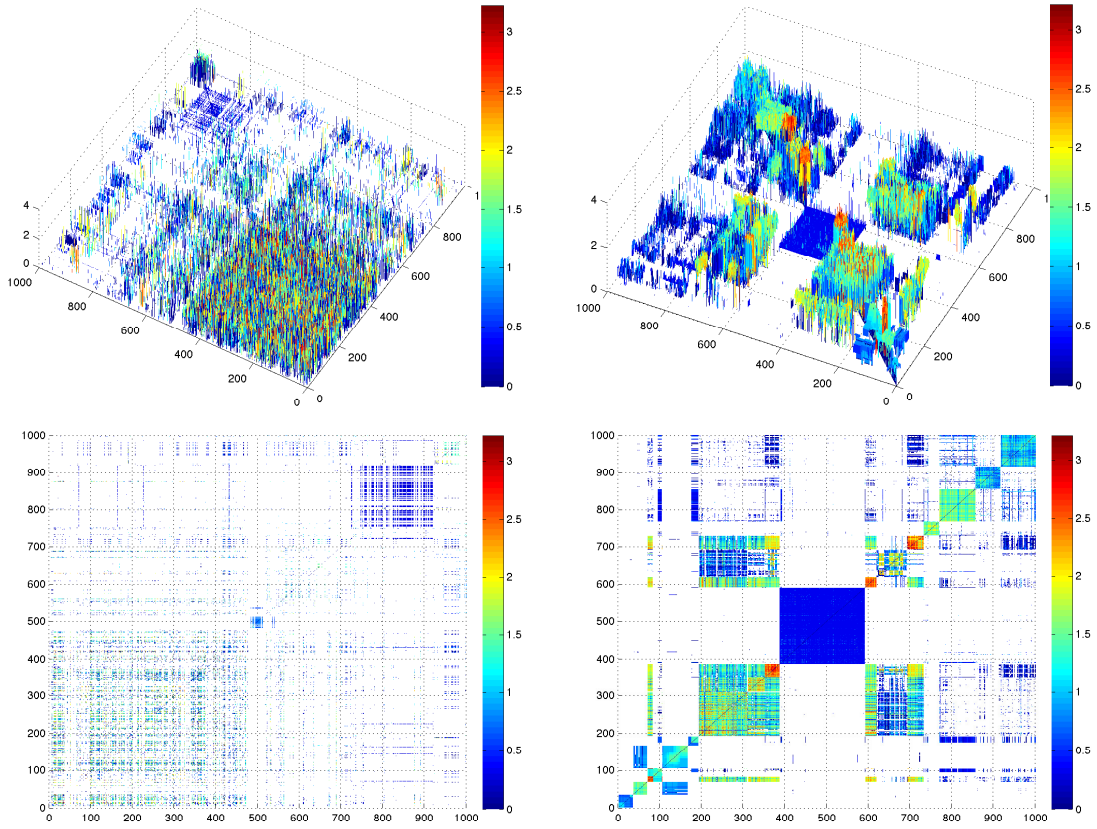
In Section 5, we describe 4 types of spammers structure: Complete Bipartite Graph, Bipartite Graph, Power law in-degree and link farm. In the experiment, we find that Twitter also has these spammer structures. To give a clear view of the difference between these spammer structures on Twitter, we plot 4 attributes: in-degree, out-



(A) Sorted by IDs

(B) Sorted by clusters

FIGURE 17: Jaccard similarities between 129 near-duplicates.



(A) Sorted by IDs

(B) Sorted by clusters

FIGURE 18: Common friends of 1K random spammers plotted in log10 scale.

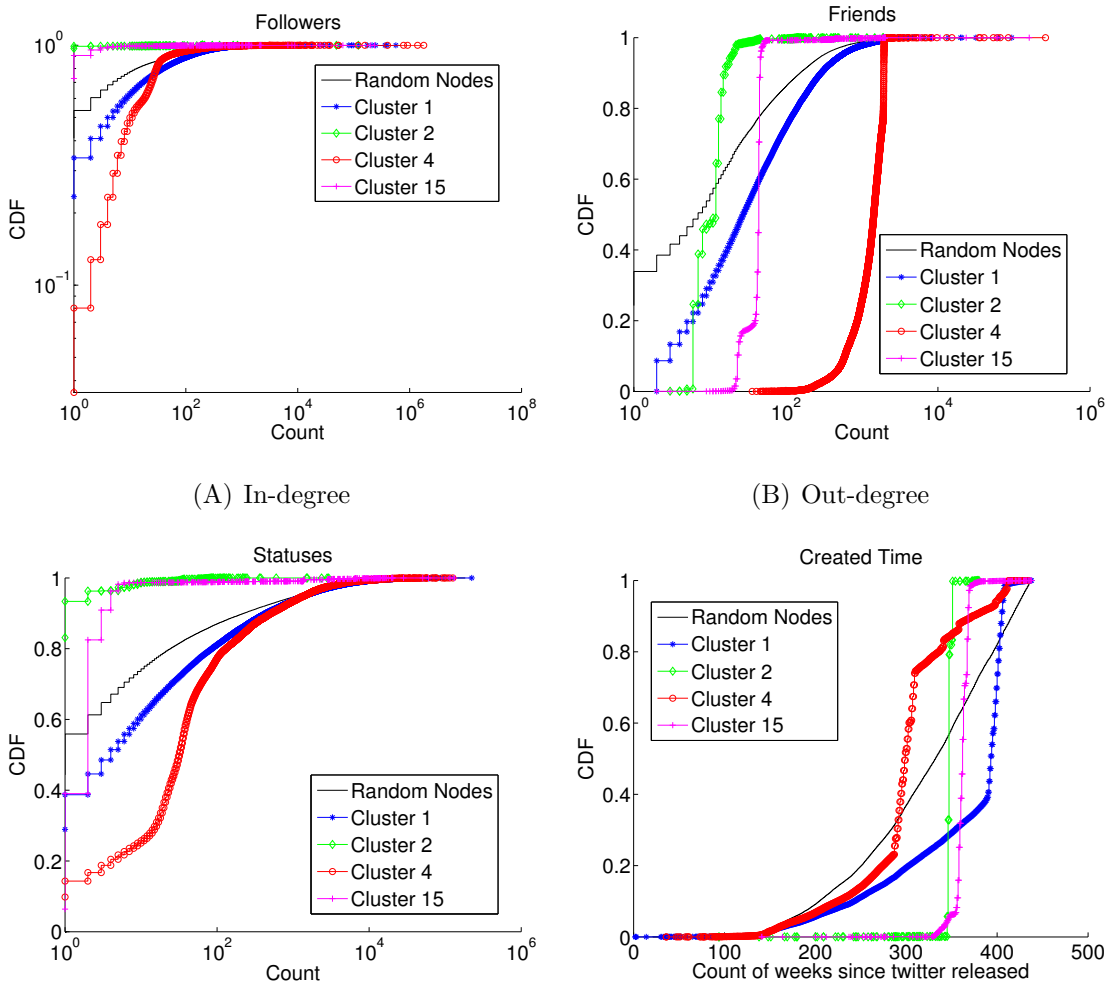
	Spammers					
Cluster	In-deg (avg)	Out-deg (avg)	# Spammers ($\times 10^6$)	#Links ($\times 10^6$)	# Near duplicates	Name
1	2.59	11.79	0.146	1.723	2	
2	13.13	52.69	0.2	10.672	25	
3	19.64	810.69	0.205	166.439	6	
4	19.81	50.63	0.313	15.855	3	
5	23.92	37.01	0.148	5.471	4	
6	57.34	724.85	0.549	398.271	9	
7	69.52	1063.23	0.308	327.082	4	
8	102.87	1372.58	0.188	258.615	2	
9	111.64	132.37	0.971	128.583	2	
10	136.03	1230.29	0.178	218.510	3	
11	201.79	775.96	0.497	385.464	59	
12	214.17	1217.68	0.173	210.656	2	
13	359.38	396.68	0.184	73.230	2	
14	501.10	789.20	0.417	328.784	2	
15	510.01	793.91	0.330	262.152	2	
16	570.05	862.54	0.474	408.594	2	
Sum			4.932	2,608.195	129	
Mean	181.31	528.87				

TABLE 6: 16 near-duplicates clusters.

Cluster	Protected	Location	Created_at	Friends_count	Followers_count	Statuses_count	Language
1	False(100.00%)	Unknown(99.67%)	2013-02(81.57%)	6(23.81%)	0(96.97%)	0(83.02%)	en(99.68%)
2	False(99.97%)	Unknown(99.63%)	2013-05(33.88%)	44(25.52%)	0(72.57%)	2(43.19%)	en(99.11%)
3	False(97.48%)	Unknown(90.62%)	2013-07(25.00%)	33(2.83%)	0(27.69%)	0(11.61%)	en(68.96%)
4	False(99.96%)	Unknown(99.72%)	2013-06(6.06%)	30(13.54%)	0(17.47%)	0(67.91%)	en(95.25%)
5	False(99.99%)	Unknown(99.83%)	2014-07(55.68%)	25(62.28%)	0(85.49%)	0(91.88%)	en(96.93%)
6	False(96.49%)	Unknown(80.14%)	2012-03(4.10%)	1920(0.32%)	1(17.42%)	0(6.24%)	ru(66.63%)
7	False(92.80%)	Unknown(73.10%)	2009-04(2.82%)	1920(0.58%)	1(9.99%)	0(10.70%)	en(59.23%)
8	False(96.50%)	Unknown(75.61%)	2012-05(13.25%)	1940(0.60%)	5(5.89%)	0(9.95%)	en(60.23%)
9	False(95.69%)	Unknown(80.33%)	2014-03(15.70%)	2(8.78%)	0(23.29%)	0(28.85%)	en(42.17%)
10	False(94.68%)	Unknown(80.17%)	2012-01(2.69%)	1920(0.80%)	1(11.20%)	0(18.17%)	en(57.38%)
11	False(96.72%)	Unknown(78.72%)	2012-03(16.13%)	10(0.48%)	0(19.22%)	0(4.30%)	en(53.15%)
12	False(95.50%)	Unknown(80.33%)	2014-03(17.72%)	1937(0.29%)	1(10.43%)	0(33.80%)	en(67.92%)
13	False(94.29%)	Unknown(70.29%)	2014-06(35.50%)	90(8.78%)	0(23.95%)	0(16.21%)	en(85.33%)
14	False(92.40%)	Unknown(48.42%)	2013-05(6.33%)	26(0.26%)	23(0.87%)	1(7.24%)	en(75.70%)
15	False(95.99%)	Unknown(62.85%)	2013-08(3.76%)	2001(1.08%)	105(0.25%)	1(2.49%)	pt(63.29%)
16	False(91.55%)	Unknown(44.57%)	2013-10(5.53%)	2001(0.49%)	40(0.45%)	1(1.39%)	en(62.45%)

TABLE 7: Most appeared attributes of 16 clusters on Twitter

degree, statuses count and creation time in Fig. 19. Each plot contains 5 groups of nodes: 4 clusters that represent these 4 structures and 1 for the random nodes.



(A) In-degree

(B) Out-degree

(C) Statuses counts

(D) Register time

FIGURE 19: Attributes of difference types of spammers and random nodes.

6 Spammers Links

First, we study spammer-to-spammer links. In a link farm, spammer will connect each other to boosts the rank of the subgraph. We define two ratios , in-loop ratio and out-loop ratio, to show the fraction that a spammer sent and received links to their degrees. The in-loop ratio of node i is defined as

$$l_{in_i} = \frac{|In\_links\_from\_spammer|}{In\_degree} \tag{1}$$

And the out-loop of node i is defined as

$$l_{out_i} = \frac{|Out\_links\_from\_spammer|}{Out\_degree} \tag{2}$$

We calculate the average in-loop ratio L_{in} and out-loop ratio L_{out} and show then in Tabletab:twitter:Link.

Next, we study the reciprocity, which is defined as the ratio of the number of links pointing in both directions to the total number of links [25]. As the twitter is a direct network, users only follow the one that interests them. The reciprocity of a user shows the follow-back ratio of that user. Here we define two reciprocities: in-reciprocity and out-reciprocity. For a node i, in-reciprocity is the ratio of the number of links pointing to this node in both directions to the in-degree of the node, which is defined as

$$r_{in_i} = \frac{|followers \cap friends|}{|friends|} \tag{3}$$

While out-reciprocity is the ratio of the number of links pointing a node in both directions to the total number of out-links, which is defined as

$$r_{out_i} = \frac{|followers \cap friends|}{|followers|} \tag{4}$$

In the experiment, we randomly select 100 spammers from each cluster and fetch out all their out links and calculate the average in-reciprocity R_{in} and out-reciprocity R_{out}, which can be found in Table 8.

We can see that some spammer groups like cluster 1 and 5 do not have any spammers as their followers, while some cluster like cluster 4, 14, and 15 have nearly 40% followers as spammers. For the out links, some spammers groups, such as cluster 2 and 9 do not follow spammers. And the spammers in cluster 3, 15 spends nearly 20% out links on spammers.

For the follow-back ratio, we take a sample of 100 random nodes and get a baseline: $R_{in_b} = 24.083\%, R_{out_b} = 11.668$. Then we calculate the average follow-back ratio

Cluster	$L_{in}(\%)$	$L_{out}(\%)$	$R_{in}(\%)$	$R_{out}(\%)$
1	0.000	13.483	1.272	0.385
2	4.000	0.000	0.000	0.000
3	9.536	18.264	14.283	0.120
4	37.472	6.123	0.143	0.043
5	0.000	11.033	2.000	0.034
6	13.223	5.492	42.697	0.722
7	4.860	4.498	37.958	0.559
8	0.914	4.311	37.269	0.522
9	2.792	0.519	40.419	15.884
10	5.780	5.308	48.524	0.735
11	4.033	15.656	25.932	0.401
12	2.951	3.363	33.556	0.477
13	1.568	14.528	38.041	9.976
14	36.795	15.809	42.197	12.815
15	40.601	19.991	75.404	31.699
16	26.619	13.514	60.216	21.618

TABLE 8: Spammer links.

of the spammers in each cluster showed in the fourth and fifth columns of Table 8. Spammer groups like 2, 4 and 5 have a extremely low follow-back ratio while some groups have a very high ratio compared to the base-line.

CHAPTER VI

Conclusion

In this thesis, we study the problem of detecting spammers in Online Social Networks. First, we take a review of the related works. We selected the works by Thomas et al. [29] and Ghosh et al. [11] which give us a comprehensive overview of the spammers features. Meanwhile, we selected 3 works [4, 28, 30] that using machine learning techniques to identify spammers among OSNs. Although the existing works have classified spammers with high accuracy, they all study the problem from the view of analysis and learn the features from the existing spammers.

In our research, we treat the spammer detection problem from a different point of view. We focus finding the spam links instead spam profile or content. We define the concept of near-duplicates in OSNs. By finding these near-duplicates, we are able to identify the spammers that following the near-duplicates.

In chapter IV, we apply our method on Sina Weibo. We successfully identified 395 near-duplicates among top 10K users, 12 million spammers and 741 million spam links. We cluster these near-duplicates and the corresponding spammers into 34 clusters and analysis the properties of each cluster. In chapter V, we apply our method on Twitter. First, we take a sample of Twitter network. By analysis the sample, we are able to estimate some global Twitter properties such as the population, top users, average in- and out-degree. Next, we apply our spammer detection method on Twitter. We find 129 near-duplicates among top 30K users, 4.93 million spammers and 2,608 million spam links, which are grouped in 16 clusters. Meanwhile, we study in and out links as well as follow back status of spammers. We summarize our finds in Table 9.

Network	Weibo	Twitter
Sample ratio	0.486%	0.086%
Near-duplicates	395	129
Clusters	34	16
Nodes	242.80 million	1.16 billion
Spammers	12 million	4.93 million
Edges	7.83 billion	83.38 billion
Spam links	741 million	2.608 billion

TABLE 9: Summary

REFERENCES

[1] (2010). Twitter phishing hack hits BBC, Guardian and cabinet minister.

[2] (2014). About Twitter, Inc. — About.

[3] Alexa.com (2014). Alexa Top 500 Global Sites.

[4] Benevenuto, F., Magno, G., Rodrigues, T., and Almeida, V. (2010). Detecting spammers on twitter. In *Collaboration, electronic messaging, anti-abuse and spam conference (CEAS)*, volume 6, page 12.

[5] Breiman, L. (2001). Random forests. *Machine learning*.

[6] Cha, M., Haddadi, H., Benevenuto, F., and Gummadi, P. (2010). Measuring User Influence in Twitter: The Million Follower Fallacy. *Proceedings of international AAAI Conference on Weblogs and Social - ICWSM ‘‘10*.

[7] Chen, C., Wu, K., Srinivasan, V., and Zhang, X. (2013). Battling the internet water army. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining - ASONAM ’13*, pages 116–120, New York, New York, USA. ACM Press.

[8] Cheng, B., Fu, J., and Huang, J. (2013). Detecting Zombie Followers in Sina Microblog based on the Number of Common Friends. *International Journal of Advancements in Computing Technology*, 5(2):612–620.

[9] Chu, Z., Gianvecchio, S., Wang, H., and Jajodia, S. (2012). Detecting Automation of Twitter Accounts: Are You a Human, Bot, or Cyborg? *IEEE Transactions on Dependable and Secure Computing*, 9(6):811–824.

[10] Fu, K.-w. and Chau, M. (2013). Reality check for the Chinese microblog space: a random sampling approach. *PloS one*, 8(3):e58356.

[11] Ghosh, S., Viswanath, B., Kooti, F., Sharma, N. K., Korlam, G., Benevenuto, F., Ganguly, N., and Gummadi, K. P. (2012). Understanding and combating link farming in the twitter social network. In *Proceedings of the 21st international conference on World Wide Web - WWW '12*, page 61, New York, New York, USA. ACM Press.

[12] Giles, J. (2011). Social-bots infiltrate Twitter and trick human users. *New Scientist*, 209(2804):28.

[13] Gjoka, M., Kurant, M., Butts, C. T., and Markopoulou, A. (2009). A Walk in Facebook: Uniform Sampling of Users in Online Social Networks.

[14] Harkreader, R. (2013). Empirical Evaluation and New Design for Fighting Evolving Twitter Spammers. *IEEE Transactions on Information Forensics and Security*, 8(8):1280–1293.

[15] Henzinger, M. (2006). Finding near-duplicate web pages. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '06*, page 284, New York, New York, USA. ACM Press.

[16] Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features.

[17] Kwak, H., Lee, C., Park, H., and Moon, S. (2010). What is Twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web - WWW '10*, page 591, New York, New York, USA. ACM Press.

[18] Lee, K., Caverlee, J., and Webb, S. (2010). Uncovering social spammers: social honeypots + machine learning. In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval - SIGIR '10*, page 435, New York, New York, USA. ACM Press.

[19] Leskovec, J. and Sosič, R. (2014). Snap.py: SNAP for Python, a general purpose network analysis and graph mining tool in Python. \url{http://snap.stanford.edu/snappy}.

[20] Liu, L. and Jia, K. (2012). Detecting Spam in Chinese Microblogs - A Study on Sina Weibo. In *2012 Eighth International Conference on Computational Intelligence and Security*, pages 578–581. IEEE.

[21] Lu, J. and Li, D. (2013). Bias Correction in a Small Sample from Big Data. *IEEE Transactions on Knowledge and Data Engineering*, 25(11):2658–2663.

[22] Morik, K., Brockhausen, P., and Joachims, T. (1999). Combining statistical learning with a knowledge-based approach: a case study in intensive care monitoring.

[23] Myers, S. A., Sharma, A., Gupta, P., and Lin, J. (2014). Information network or social network?: the structure of the twitter follow graph. In *23rd International World Wide Web Conference, {WWW} '14, Seoul, Republic of Korea, April 7-11, 2014, Companion Volume*, pages 493–498. International World Wide Web Conferences Steering Committee.

[24] Newman, M. (2010). *Networks: an introduction*. Oxford University Press.

[25] Newman, M., Forrest, S., and Balthrop, J. (2002). Email networks and the spread of computer viruses. *Physical Review E*, 66(3):035101.

[26] Sarver, R. (2011). Update on Whitelisting.

[27] Song, J., Lee, S., and Kim, J. (2011). Spam filtering in twitter using sender-receiver relationship. *14th International Symposium, RAID 2011, Menlo Park, CA, USA, September 20-21, 2011. Proceedings*.

[28] Stringhini, G., Kruegel, C., and Vigna, G. (2010). Detecting spammers on social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference on - ACSAC '10*, page 1, New York, New York, USA. ACM Press.

[29] Thomas, K., Grier, C., Song, D., and Paxson, V. (2011). Suspended accounts in retrospect: an analysis of twitter spam. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference - IMC '11*, page 243, New York, New York, USA. ACM Press.

[30] Wang, A. H. (2010). Don't follow me: Spam detection in Twitter. *Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on*, pages 1–10.

[31] Wang, H. and Lu, J. (2013). Detect inflated follower numbers in OSN using star sampling. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining - ASONAM '13*, pages 127–133, New York, New York, USA. ACM Press.

[32] Weng, J., Lim, E.-P., Jiang, J., and He, Q. (2010). TwitterRank: finding topic-sensitive influential twitterers. In *Proceedings of the third ACM international conference on Web search and data mining - WSDM '10*, page 261, New York, New York, USA. ACM Press.

[33] Wu, B., Goel, V., and Davison, B. (2006). Propagating Trust and Distrust to Demote Web Spam. *MTW*.

[34] Yang, C., Harkreader, R., and Gu, G. (2011). Die free or live hard? empirical evaluation and new design for fighting evolving twitter spammers. *Recent Advances in Intrusion Detection*.

[35] Yang, Z., Wilson, C., Wang, X., Gao, T., Zhao, B. Y., and Dai, Y. (2014). Uncovering social network Sybils in the wild. *ACM Transactions on Knowledge Discovery from Data*, 8(1):1–29.

[36] Zhou, J., Li, Y., Adhikari, V. K., and Zhang, Z.-L. (2011). Counting YouTube videos via random prefix sampling. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference - IMC '11*, page 371, New York, New York, USA. ACM Press.

[37] Zhou, Y., Chen, K., Song, L., Yang, X., and He, J. (2012). Feature Analysis of Spammers in Social Networks with Active Honeypots: A Case Study of Chinese Microblogging Networks. In *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 728–729. IEEE.

VITA AUCTORIS

NAME:	Yi Zhang
PLACE OF BIRTH:	Hancheng, Shaanxi province, China
YEAR OF BIRTH:	1989
EDUCATION:	Xidan University, B.Eng., Computer Science and Technology, Xi'an, China, 2012
	University of Windsor, M.Sc in Computer Science, Windsor, Ontario, 2015