PLAYING HIDE-AND-SEEK WITH SPAMMERS:DETECTING EVASIVE ADVERSARIES IN THE ONLINE SOCIAL NETWORK DOMAIN

A Thesis

by

ROBERT CHANDLER HARKREADER

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 2012

Major Subject: Computer Science

PLAYING HIDE-AND-SEEK WITH SPAMMERS: DETECTING EVASIVE ADVERSARIES IN THE ONLINE SOCIAL NETWORK DOMAIN

A Thesis

by

ROBERT CHANDLER HARKREADER

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Guofei Gu
Committee Members,	James Caverlee
	Alexander Parlos
Department Head,	Duncan M. Walker

August 2012

Major Subject: Computer Science

ABSTRACT

Playing Hide-and-Seek with Spammers: Detecting Evasive Adversaries in the Online Social Network Domain. (August 2012)

Robert Chandler Harkreader, B.S., Texas A&M University

Chair of Advisory Committee: Guofei Gu

Online Social Networks (OSNs) have seen an enormous boost in popularity in recent years. Along with this popularity has come tribulations such as privacy concerns, spam, phishing and malware. Many recent works have focused on automatically detecting these unwanted behaviors in OSNs so that they may be removed. These works have developed state-of-the-art detection schemes that use machine learning techniques to automatically classify OSN accounts as spam or non-spam. In this work, these detection schemes are recreated and tested on new data. Through this analysis, it is clear that spammers are beginning to evade even these detectors. The evasion tactics used by spammers are identified and analyzed. Then a new detection scheme is built upon the previous ones that is robust against these evasion tactics. Next, the difficulty of evasion of the existing detectors and the new detector are formalized and compared. This work builds a foundation for future researchers to build on so that those who would like to protect innocent internet users from spam and malicious content can overcome the advances of those that would prey on these users for a meager dollar.

To my mom and dad who always believed in me.

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Guofei Gu, my committee members and my friends in the SUCCESS lab.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Online social networking sites (OSNs) are websites in which users can create profiles, establish connections with friends and traverse these connections[1]. OSNs have existed since the creation of *sixdegrees.com* in 1997. Since then, hundreds of OSNs have been created for a variety of purposes. OSNs did not become mainstream until the creation of websites such as MySpace, LinkedIn, Facebook and Twitter. Facebook is the largest of the mainstream OSNs, boasting over 500 million users [2]. The popularity of these websites has thrust online social networking into the spot light. This has attracted news media, celebrities and unfortunately, spammers. A paradigm example of the extraordinary role that OSNs play in our society was when OSNs such as Facebook and Twitter played critical roles in setting up protests in Cairo, Egypt in 2011 [3].

The battle to protect OSNs from spam has waged for several years now. In August of 2009, nearly 11 percent of all Twitter posts were spam [4]. Annoying advertisements are not the only concern for users on OSNs. Spammers have also used this new platform for spreading malware, luring users to phishing websites and even hosting botnet command and control channels [5]. There have also been several reports of attacks on Twitter. The infamous Acai Berry attack forced users to post an advertisement to all of their followers about the supposed health benefits of Acai berries[6]. Koobface is a worm that propgates

---

This thesis follows the style of IEEE Transactions on Knowledge and Data Engineering.

itself through several different OSNs [7]. This worm was first detected in 2008, but was easily thwarted by updates to the OSNs. Now a more evolved version is propagating itself through OSNs once again [8]. Spam has a direct affect on the OSN company itself because it costs money to store and maintain all of these spam accounts and posts. In the bigger picture, when these spam attacks steal identities from the members of society they not only cost those families and credit card agencies thousands of dollars but also cause people to feel less secure on the Internet [9]. These feelings of insecurity make people feel uneasy about shopping online, which costs all online businesses money.

Due to these threats, many OSNs and even several research labs have put effort into stopping this behavior. For instance, Twitter has published a set of rules that users must abide by, called The Twitter Rules [10]. These rules define behaviors that could cause an account to be considered as a spam account. As defined by The Twitter Rules, spam-like behavior includes following many accounts without having many users follow your account. This is known as having a high *following-follower ratio*. Another spam-like behavior posting duplicate tweets. If Twitter deems an account to be a spam account, the account will be suspended from Twitter. Twitter also has a method for users to report other accounts as spam, however, this relies on action from the average Twitter user. In academia, works such as [11], [12], [13], [14] and [15] focus on using supervised learning techniques to classify spam accounts from non-spam accounts.

These methods have classified spammers with high accuracy with low numbers of false positives. Twitter has been able to greatly reduce the amount of

spam on its website. However, as we will show, there is still quite a bit of spam on Twitter. The reason for this is the adaptability of spammers. Many spammers have already adapted to these techniques and have been able to evade them. For instance, many websites have been set-up for the sole purpose of purchasing followers on Twitter. This is a direct evasion of the rule from The Twitter Rules which says you should not have a high *following-follower ratio* [16]. There are also tools available to help you modify your tweets without changing the meaning [17]. These evasive tactics require a response from the research community to design more robust schemes that are more difficult to evade than existing works.

This work performs an in-depth analysis of these advanced spammers, referred to as evasive spammers. First this work reproduces existing detection schemes and evaluates them on data containing evasive spammers. Next, an extensive analysis is performed on the evasive spammers to determine how they are able to evade these detection schemes. Based on this analysis, a similar, yet more robust detection scheme is proposed that is able to detect spammers that successfully evaded previous work. Additionally, we quantify the robustness of each feature of our detection scheme as well as the features of previous detection schemes.

In summary, the contributions of this paper are as follows:

- Three state-of-the-art detection schemes are reproduced and analyzed using a new data set.
- Evasive spammers are discovered and an in-depth analysis of their behavior is performed.

- A similar, but more robust, detection scheme is designed and implemented. This detection scheme is able to correctly classify 13% more spammers than the best existing detector while maintaining the same false positive rate.

- The robustness of each detection scheme is analyzed and quantified.

CHAPTER II

LITERATURE REVIEW

As OSNs have become more popular in the world, many researchers have turned their attentions to the study of these online communities. Kwa *et al.* [18] has performed a study of the behavior of accounts on Twitter. This includes a comprehensive and quantitative study of the distributions of various statistics such as number of followers, number of followings as well as reciprocity between accounts. Cha *et al.* [11] performed an in-depth measurement study of the accounts on Twitter.

Since spam is such a real concern for many companies in many industries, the goal of spam reduction is a hot topic. OSNs such as Twitter have directly attacked spam by publicizing defining characteristics of spammers in The Twitter Rules [10] as well as the creation of the Spammer Reporting tool [19]. Third party vendors have also created applications in order to thwart the onslaught of spam on Twitter. Blocky has created a blacklist of Twitter accounts based on votes from Twitter users [20]. Tweet Blocker attempts to assign a score to each Twitter account based on basic information such as registration time and *following-follower ratio*. This helps other Twitter users know how credible a Twitter account is before following them.

There are also several research publications about spam in OSNs. Koutrika *et al.* [21] proposed a technique to detect tag spam in tagging systems. This spam would direct users searching for material on a particular topic to spam instead. They rank the credibility of a tag based on a tagger's reliability, which prevents spammers from performing this attack. Benevenuto *et al.* [22, 23] uses

a supervised learning technique to classify videos posted by spammers from normal videos on YouTube. Gao *et al.* [24] identifies and studies campaigns on OSNs, identified based on the spam URLs that they post.

Even more related to this work, there have been several publications about detecting spammers in OSNs, including Twitter. Several works use supervised learning techniques to classify spammers from normal users on OSNs [12], [13], [14], [15]. These works first create training data by labeling a set of known spam accounts and a set of known non-spam accounts. Next they extract features from these accounts that will help the algorithm distinguish between normal and spam accounts. Lee *et al.* deployed social honeypot accounts in MySpace as well as Twitter to collect spam accounts. They then used features such as URLs per post, replies per post and number of connections to classify accounts. Benevenuto *et al.* also used supervised learning techniques to detect spammers on Twitter with features such as tweet contents, number of hashtags per post and number of followings and followers. Wang [14] used unique features such as a novel reputation score and number of duplicate Tweets as well as similar features. These works all create an extensive examination of the use of supervised learning in detection of spammers on OSNs using basic features.

This work recreates the best performing of these previous works and evaluates them on a new data set, with a different type of OSN spammer. The analysis presented about these previous works shows that they are vulnerable to evasion tactics that are already being used in the wild. This work performs an in-depth analysis of these tactics in order to develop new, more robust features that are able to detect these evasive spammers. Also, this work presents

a quantitative study of the robustness of spam detection features. Thus, this work adds value to the research community and proposes a new area of concern: the evasive spammer.

CHAPTER III

OVERVIEW

This chapter will describe the experimental domain, Twitter, and formally define key components of the problem. Then this chapter will formalize the problem statement.

A. Description of Twitter

Twitter is the fastest growing OSN on the Internet today. There are already over 200 million users and many more are joining each day [25]. Twitter is a micro-blogging website. A blog is a weblog which is a post hosted on the Internet, accessible to others. Typically these weblogs can be as long or as short as the author would like them to be. A micro-blog, however, limits the number of characters the author can write in one blog post. In the case of Twitter, authors can only post 140 characters at a time. This restriction, while seemingly a hindrance to say what you really want to say, in fact produces an interesting result. Since authors are forced to be more concise, they tend to post more often and it doesn't take much reading to understand their point. This in-turn creates a real-time stream of the thoughts and feelings of the entire world, pouring out for everyone to see. In a sense, it is like the pulse of the world at any Twitter user's fingertips.

However, it is impossible to read every single post, so one must make use of the features provided by Twitter in order to sort out the madness. One way to do so is to choose a select group of people that you would like to hear from.

These may be your friends, co-workers, favorite celebrities or even your favorite researchers. In order to do this, you search for their account on Twitter and then click the button to follow them. Once you click this button, you are able to see all of the posts of the users you follow each time you log in to Twitter. Also, when one Twitter user follows another, the user being followed receives an e-mail alert telling him that he has been followed, including a link to the follower's Twitter page.

Since Twitter is an OSN, there is a social aspect to it as well. If you would like to communicate with someone, there are two options. The first is the *mention*. By placing a token in your Tweet with the '@' symbol, followed by the screen name of another user on Twitter, one can mention another user in their Tweet. When one user mentions another, the user being mentioned can see this Tweet from their account *even if the user is not following the user that mentioned him*. This means that the mentioned user will be notified of the Tweet of the user that mentioned him, without giving any consent in the matter.

Another interesting aspect of Twitter is the ability to see the opinions of other Twitter users on a particular topic or current event. *Trending topics* can assist a Twitter user in this endeavor. Trending topics are the recent most frequently tweeted phrases. This could be a celebrities name, such as "Justin Bieber" or a current event such as "Japanese earthquake". Twitter automatically extracts these trending topics from their public timeline. Twitter provides a list of currently trending topics as links once a user logs in to the website. A user can click the "Justin Bieber" link to see what thousands of people are

thinking about Justin Bieber right this second. Also, *hashtags* are used for a similar purpose. A hashtag is a token in the post that begins with the '#' symbol. If a user on Twitter searches for a hashtag, he can see all the currently posted Tweets containing these hashtags. This method is used for less popular events, for example a small event happening on campus or a popular event, such as a Starcraft 2 e-sports event that is not quite popular enough to create a trending topic. These techniques are important to understand because spammers use this to their advantage.

B. Definition of a Spammer

Most people that have used the Internet have encountered spam of some kind, whether it be e-mail spam, forum spam, instant messenger spam or other types. In this chapter, a formal definition of OSN spam is given with regards to this work. Also, the motivations and typical behaviors of OSN spam are described.

Since this work focuses on the detection of spam in OSNs, particularly Twitter, The Twitter Rules are consulted to assist with formalizing the definition of a spammer. In The Twitter Rules, spam and abusive behavior are defined together in one category. This makes sense, because most abusers use spam as a method for increasing the number of users they can deceive. The Twitter Rules has many identifying behaviors for spammers. This work focuses on those accounts that "publish or link to malicious content intended to damage or disrupt another users browser or computer or to compromise a users privacy" as well as those that "use the Twitter service for the purpose of spamming anyone"[10]. Since spamming is a loosely defined term, Twitter states

that the behaviors considered to be spamming behaviors will evolve over time as spammers develop new techniques. In this work, we define a spammer to be a Twitter account that publicizes links to malicious content with intent to harm another user or publicizes spam with the intent to force unsolicited advertisements upon another user. Admittedly, this definition relies upon knowing the intentions of a particular Twitter user, which is difficult if not impossible to know with certainty. For this reason, manual verification is required to be confident of the correct classification of a particular Twitter account. The specific methodology for manual classification will be described in Chapter IV. If a concrete definition of a spammer existed, detection would be trivial and this work would not be interesting.

C. Motivation of Spammers

Regardless of the domain or behavior of a particular spammer, the motivation is the same: to earn money. By spamming links to simple advertisements, spammers can get paid for each user that clicks on the link. More devious spammers may set up scams in order to trick naïve users into giving them their money for little or nothing in return. Other malicious spammers may use Twitter as a catalyst for spreading existing malware and phishing campaigns. By infecting users on Twitter with their malware, they can sell their computers as bots in part of a botnet or steal and sell the user's private information from their computer. By directing Twitter users to phishing websites, they can steal private information and sell this information on the black market.

D. Typical Behavior of an OSN Spammer

In order to understand the behavior of OSN spammers, one must first understand how to make money on OSNs. The previous section mentioned that spammers will post links to websites with different nefarious goals. However, the key to making money through these techniques is to deceive as many users as possible into clicking these links.

Since OSNs are relatively new, the techniques of spammers are still rapidly evolving. Despite the recentness of spam on OSNs, many spammers have developed cutting edge techniques in order to get more users to click their malicious links. In order to get 10s of users to click a link, one must first expose this link to 1,000s of users. On Twitter, each post an account makes is public to anyone that can access Twitter. However, simply making the post available does not mean that many people will see the post.

There are several methods that spammers use to get users to view their posts. The most common is following. As mentioned before, when a user is followed, they are alerted by Twitter about this event. The natural reaction is to investigate this account to see if it is interesting and worth following back. When a spam account follows a legitimate account, the legitimate account is then exposed to the annoying and possibly harmful spam.

Other techniques take advantage of the Twitter specific features described in Section A. The first of which is the mention. When a spammer mentions a victim, that victim will see the mention and be exposed to that tweet. This motivates the user to investigate the account that has mentioned them to see what they had to say and if a reply may be needed. It may also encourage the

user to click any links in the post to see if the website they link to has any content that would be interesting to the user.

Another common technique used by spammers to increase exposure to their malicious links is to use *trending topics* and *hashtags*. Note that the previous attack vectors were directed on individual accounts. By posting malicious URLs in tweets containing trending topics or popular hashtags, the spammer is able to expose anyone tracking these topics to their malicious content. However, with this high reward potential also comes high risk, because polluting a trending topic or popular hashtag may frustrate legitimate users and encourage them to report the spam to Twitter.

E. Problem Statement

As mentioned before, spam causes headaches and financial loss. OSN spam takes advantage of the trust that users put into OSNs. This work and related works have focused on eliminating spam in OSNs. Specifically, the problem that this thesis investigates is the detection of advanced spammers that have evolved to avoid naive detection schemes.

In order to solve this problem, this work reproduces three state-of-the-art detection schemes and analyzes their performance on a new data set. Next, this work identifies evasive spammers and performs an in-depth analysis of their evasive tactics. Then, a similar, but more robust, detection scheme is designed and implemented. Finally, the robustness of each detection scheme is analyzed and quantified.

CHAPTER IV

REPRODUCING CURRENT DETECTION SCHEMES

In this chapter, the current state-of-the-art detection schemes are reproduced and tested with new data. This chapter describes the previous detection schemes in-depth and also describes how the new data set was collected.

A. Description of Current Methods

This section describes the way that current detection schemes work.

1. The Machine Learning Technique

The state-of-the-art detection schemes use the machine learning technique. This technique essentially "trains" a program to know the differences between a spam account and a normal account. These programs must be trained on labeled data, then they can be used to automatically classify unidentified accounts.

a. Feature Vector Representation

First, the machine learning program needs a way to comprehend these accounts. To do this, features are extracted from each account to describe these accounts. These features form a feature vector representation of each account. This makes it possible for the program to learn what values of each feature are typical for spam accounts and for normal accounts.

b. Labeled Data

In order to train the program and evaluate the detection scheme, known spam accounts and known normal accounts are needed. These known spam and normal accounts are used to train the program. Then, this program can be used to automatically classify accounts. Also, part of the known spam and normal accounts can be used to train the program and the rest can be used to evaluate the trained program. This technique is known as cross-validation.

2. Features

This section describes some of the features that are used by the current detection schemes. These features are divided into two categories: relational features and content-based features.

a. Relational Features

One of the techniques that spammers use to gain attention, mentioned in Chapter III, is following target accounts. As was mentioned, this behavior creates a high following-follower ratio. Thus, one of the relational features is the following-follower ratio. The higher this ratio is, the more likely a user is to be a spammer. However, there can be false positives, for example, when new legitimate accounts join Twitter and follow some of their favorite celebrities with out having many followers. For this reason, the number of followers and the number of followings are also features. A slightly more robust relational feature is the number of bidirectional connections an account has. This means the account follows a user and that same user follows the account back. Spammers follow

many users and typically these users are uninterested in following these spam accounts back. Thus, the fewer bidirectional connections an account has, the more likely it is to be spam.

Another relational feature is the number of bi-directional links. A bi-directional link occurs between two Twitter accounts when they follow each other. This indicates that both parties are interested in each other and is a stronger relationship. The number of bi-directional links an account has reflects the reciprocity between an account and the users that it follows. Since Twitter spammers usually follow a large number of legitimate accounts and few of them follow the spam accounts back, the reciprocity of the spammers is lower than that of legitimate accounts. Thus, this feature has been used to detect spammers by existing work.

b. Content Based Features

Also described in Chapter III, Twitter spammers almost always post URLs in their spam tweets in order to direct a user to a website that will try to harm or deceive the user. For this reason, the number of URLs posted per tweet is a content based feature. Since spammers also abuse features available to legitimate Twitter users such as the mention and the hashtag, described in Section A. Thus, the number of hashtags and mentions per tweet have also been used as content based features. Since the beginning of the battle between spammers and anti-spammers, common spam terms have been used to identify e-mails or forum posts that are spam. Previous works have also attempted to use a list of known spam terms to combat spam on Twitter. Also, it is

beneficial for a spammer to post several tweets containing the same URL in order to increase traffic to the website. For this reason, another content based feature is the number of pairs of duplicate tweets. Since it is easy for a spammer to slightly modify a tweet, some works have used a more robust version of this feature, which is tweet similarity. This calculates the tweet similarity of all tweets and assigns a score based on how similar the tweets are with each other.

The remaining features used by these existing techniques are enumerated in Chapter VII. Since all detection schemes use the machine learning technique, each detection scheme can be described by the features they choose to extract from the accounts.

B. Data Collection

Before analyzing these existing methods, new labeled data must be collected for training and testing. In this section, the methods for collecting data from Twitter are described. First, a large sample of data is taken from Twitter using the Twitter API [26]. Then, this data is analyzed using various methods in order to find and label spammers to be used to train the previous detection schemes.

1. Crawling Twitter

Typical graph traversing algorithms, such as depth first search (DFS) and breadth first search (BFS) tend to create sampling biases when only crawling a portion of a given graph [27]. To avoid such a bias, a new graph traversal method is used. First, 20 seed accounts are gathered from Twitter's public time-

line. Using Twitter API, each access to Twitter's public timeline returns the 20 most recent tweets [28]. The author account of each tweet is then used as a seed account. For each of these 20 seed accounts, their followers and followings are also crawled. For each crawled account, all data available from Twitter is stored in a database. Then a new set of seed users are obtained from the public timeline. This reduces the amount of bias since the randomness of the public timeline ensures that not all of the crawled users come from the neighborhood of one particularly popular user.

This crawl resulted in the collection of information from nearly 500,000 users. Table I shows the details on how many users and tweets were crawled. The spammer identification method used, which will be explained further in the next section, relies heavily on the URLs that users put in their tweets. For this reason the URLs were extracted from the users' tweets. Typically users of Twitter will use URL shortening tools to shrink the number of characters that make up a URL. This is due to the 140 character per tweet limitation. These tools create a webpage that has no content and does nothing more than redirect to the destination web page. A spammer that is trying to hide the actual URL might use several redirections, creating a redirection chain. Since the spammer identification method relies on having the final URL, a URL redirection tracing technique tracks these redirection chains to the final URL.

2. Identifying Spam Accounts

In this section, the method for identifying spammers in the dataset for the labeled data will be detailed. There are several different types of spammers.

Table I. Twitter Crawl Summary

Name	Value
Number of Twitter accounts	485,721
Number of Followings	791,648,649
Number of Followers	855,772,191
Number of tweets	14,401,157
Number of URLs Extracted	5,805,351

This work focuses on spammers that post links to malicious websites such as phishing or malware hosting sites.

a. Identifying Suspect Accounts

To discover these spammers, two blacklists, Google Safe Browsing [29] and PhishTank [30], are used along with Capture-HPC, a high-interaction honeypot client to identify websites that are either hosting malware or phishing for private information. Each tweet containing at least one of these malicious URLs is considered to be a *spam tweet*. The ratio of spam tweets to normal tweets from a particular account is defined as the *spam tweet ratio*. Since non-malicious Twitter users may accidentally post a link to a website that is hosting malware or a phishing website, this work focuses on those that have a high spam tweet ratio. Accounts that have a spam tweet ratio higher than 10% are considered *suspect accounts*. Any accounts that are not suspect accounts are considered normal users and may be used as labeled data in the machine learning algorithm. There were 2,933 suspect accounts found in this dataset which were then subject to manual verification.

b. Manual Verification

To manually verify whether or not an account is spam, the definition of spam from Section B is used. Recall that this definition is rather loose in that it relies on understanding the intentions of a Twitter account. Since this work relies on the accounts labeled as spam to indeed be spam, the verifiers were asked to assume each account was innocent unless their malicious intentions were obvious. The verifiers attempt to judge whether the account is trying to deceive users into violating The Twitter Rules or giving up their money or personal information. There are a few scenarios where a non-malicious account would post many malicious links: 1) These links have been incorrectly labeled by the blacklist; 2) The account incorrectly believes these links are non-malicious; 3) The account posted links to a non-malicious website which then became compromised. If any of these seem to be the case, then the intentions of the account are judged to be benign.

The manual verification process involves three different parties that judge each suspect account. If the first two manual verifiers disagree on the intention of a particular account, the third makes the final judgment. This method minimizes human error, however, it does not eliminate it. While the use of erroneous data is undesirable, it is acceptable due to the large dataset. All spam detection systems face the problem of a noisy dataset and therefore the algorithms must be robust enough to overcome this difficulty. Also, there may be other malicious accounts in the dataset that were not found. This is due to the fact that this work focuses on a particular type of spammer. Also, blacklists are not perfect and may miss some malicious URLs. For this reason, the number

of spammers identified is a lower bound on the total number of spammers in the dataset. However, even with this subset of spammers, this work can show that they are using evasion tactics to avoid detection. From the 2,933 suspect accounts 2,060 were verified to be spammers.

First, nearly 500,000 users were crawled from Twitter. These accounts' tweets were analyzed and those accounts that post a high percentage of black-listed URLs are considered suspect accounts. From the nearly 500,000 crawled users, 2,933 were considered to be suspect accounts and from those 2,060 were manually verified as spam accounts. Next this labeled data is used to train and evaluate the existing detection schemes.

C. Testing Detection Schemes

In this section, the methodology for testing the detection schemes is described. These detection schemes are being tested to find evasive spammers. Since these schemes use the machine learning technique, the labeled data used, the features used and the machine learning algorithm will be described.

1. Labeled Data

This test uses 500 of the labeled spam accounts as well as 5000 non-spam accounts. The non-spam accounts are collected from those accounts that did not post any malicious URLs. There may be some undetected spam accounts in the 5000 non-spam accounts, however, one must expect to work with a noisy data set in real world situations.

2. Feature Extraction

Each existing detection scheme is described in previous works. For each of the 5,500 accounts the features listed in these works were extracted from the crawled data set. These features are enumerated in Chapter VII.

3. Machine Learning Algorithm

Each existing detection scheme used various machine learning algorithms and some compared the performance of several. To make a fair comparison, the same machine learning algorithm is used to evaluate each feature set. The decision tree algorithm is a commonly used algorithm and it also allows for easier analysis since the model created by the algorithm is human-readable. For these reasons, the J48 decision tree algorithm [31] is used to analyze the existing feature sets.

In order to evaluate the feature sets, 10 fold cross validation is used to find which spammers are incorrectly classified as non-spammers. A spammer that is incorrectly classified as a non-spammer is an evasive spammer. Each feature set has its own set of evasive spammers and each group is analyzed separately to discover evasion tactics. The evaluation of these detection schemes will be shown in detail along with the evaluation of the new detection scheme in Chapter VIII.

CHAPTER V

ANALYZING EVASION TACTICS

This chapter will discuss the ways in which spammers have begun to evade existing detection schemes. The previous chapter showed that existing techniques have been able to detect spammers with high accuracy, however, new data shows that new spammers are taking actions to evade these existing detection schemes. This chapter uses a set of users called *evasive spammers*. These are a set of spammers, identified in the previous chapter, that have evaded existing detection schemes. That is, these spammers were falsely classified as normal by existing detection algorithms. There are three different sets of evasive spammers, one from each existing method.

A. Description of Evasion Tactics

The main evasion tactics are utilized by the spammers to evade existing detection approaches and can be categorized into the following two types: profile-based feature evasion tactics and content-based feature evasion tactics.

1. Profile-based Evasion

A common intuition for discovering Twitter spam accounts can originate from accounts' basic profile information such as number of followers and number of tweets, since these indicators usually reflect Twitter accounts' reputation. To evade such profile-based detection features, spammers mainly utilize tactics including gaining more followers and posting more tweets.

Gaining More Followers: In general, the number of a Twitter account's followers reflects its popularity and credibility. A higher number of followers of an account commonly implies that more users trust this account and would like to receive the information from it. Thus, many profile-based detection features such as *number of followers*, *fofo ratio* (The ratio of the number of an account's following to its followers.) [12, 15] and *reputation score* [14] are built based on this number. To evade these features or break Twitter's 2,000 Following Limit Policy (According to this policy, if the number of followings of an account exceeds 2,000, this number must not be much more than the number of the account's followers.) [32], spammers can mainly adopt the following strategies to gain more followers. The first strategy is to purchase followers from websites. These websites charge a fee and then use an arsenal of Twitter accounts to follow their customers. The specific methods of providing these accounts may differ from site to site. The second strategy is to exchange followers with other users. This method is usually assisted by a third party website. These sites use existing customers' accounts to follow new customers' accounts. Since this method does only require Twitter accounts to follow several other accounts to gain more followers without any payment, Twitter spammers can get around the referral clause by creating more fraudulent accounts. In addition, Twitter spammers can gain followers for their accounts by using their own created fake accounts. Spammers will create a bunch of fake accounts then follow their spam accounts with these fake accounts.

Posting More Tweets: Similar to the number of an account's followers, an account's tweet number usually reflects how much this account has con-

tributed to the whole online social platform. A higher tweet number of an account usually implies that this account is more active and willing to share information with others. Thus, this feature is also widely used in the existing Twitter spammers detection approaches (e.g. [15]). To evade this feature, spammers can post more Tweets to behave more like legitimate accounts, especially recurring to utilizing some public tweeting tools or software [33].

2. Content-based Evasion

Another common indicator for distinguishing spam accounts is the content of a suspect account's tweets. As discussed in Chapter III, a majority of spam accounts make profit by alluring legitimate users to click the malicious URLs posted in their spam tweets. Those malicious URLs direct users to websites that cause harm to their computers or scam them out of their money. Thus, the percentage of tweets containing URLs is an effective indicator of spam accounts, which is utilized in works such as [12, 15, 14]. In addition, since many spammers post the same or similar malicious tweets in order to increase visibility, their published tweets show strong homogeneous characteristics. In this way, many existing approaches design content-based features such as *tweet similarity* [12, 15] and *duplicate tweet count* [14] to detect spam accounts. To evade such content-based detection features, spammers use tactics such as mixing normal tweets and posting heterogeneous tweets.

Mixing Normal Tweets: Spammers can utilize this tactic to evade content-based features such as *URL ratio*, *unique URL ratio* and *hashtag ratio* [12, 14]. These normal tweets without malicious URLs may be hand-crafted

or obtained from arbitrary users' tweets or consist of meaningless words. By mixing such normal tweets, spam accounts are able to dilute their spam tweets and make it more difficult for a detector to distinguish them from legitimate accounts.

Posting Heterogeneous Tweets: Spammers can post heterogeneous tweets to evade the content-based features such as *tweet similarity* and *duplicate tweet count*. Specifically, in this tactic, spammers post tweets with the same semantic meaning but with different terms. In this way, not only can the spammers maintain the same semantic meanings to allure victims, but also they can make their tweets variational enough to not be caught by detectors that rely on such content-based features. In fact, many public tools, e.g. Spinbot [17], can help spammers to spin a few different spam tweets into hundreds of variable tweets with the same semantic meaning but different words.

B. Validation of Evasion Scenarios

In this section, we aim to validate the four evasion tactics described in the previous section by showing real case studies and public services/tools that can be utilized by the spammers. We also implement existing detection schemes [12, 15, 14] and evaluate them on our collected examination data set. By analyzing the spammers missed (false negatives) by these works, we can show that many spammers have indeed evolved to behave like legitimate accounts to evade existing detection features.

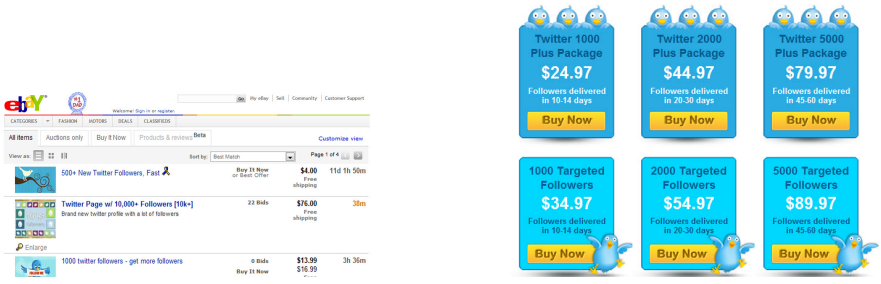
Gaining More Followers: As described in previously in this chapter, spammers can gain more followers by purchasing them, exchanging them and

creating fake accounts. In fact, several public websites allow for the direct purchase of followers. The rates per follower for each website vary. Table II shows that followers can be purchased for small amounts of money on several different websites, even including the online bidding website – Ebay, which can be seen in Fig. 1(a).

Table II. Price of Online Follower Trading

Website	Price Per Follower
BuyTwitterFriends.com	$0.0049
TweetSourcer.com	$0.0060
UnlimitedTwitterFollowers.com	$0.0074
Twitter1k.com	$0.0209
SocialKik.com	$0.0150
USocial.net	$0.0440
Tweetcha.com	$0.0470
PurchaseTwitterFollowers.com	$0.0490

Also, Fig. 1(b) shows a real online website from which users can directly buy followers. From this figure, we can find that, spammers can buy followers at a very cheap price. The website also claims that the user can buy targeted followers with specific keywords in their tweets.



(a) Bidding followers from Ebay (b) Purchasing followers from website

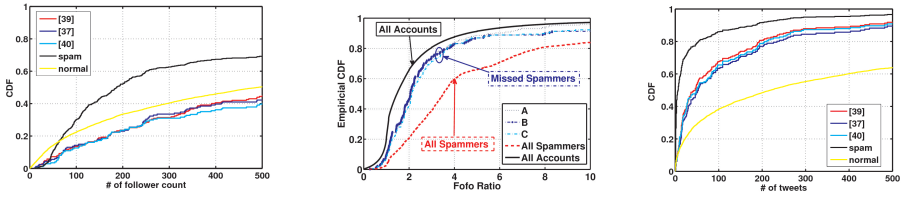
Fig. 1. Online Twitter Follower Trading Website

After showing these online services through which spammers can obtain more followers, we examine the detection features of *number of followers* and

fofo ratio from three existing work on our collected dataset. In particular, we draw the distribution of these two metrics of three sets of accounts: missed spammers (false negatives) in each of three existing approaches [12, 15, 14], *all accounts* (around 500,000 collected accounts), and *all spammers* (2,060 identified spammers). To better show the results, we label the results from [14] as A, [12] as B and [15] as C. From Fig. 2(a) and 2(b), we can see that the distributions of these two indicators of those missed spammers by the existing approaches are more similar to that of *all accounts* than that of *all spammers*. This observation shows that many spammers pretend to be more legitimate by gaining more followers.

Posting More Tweets: Besides using the web to post tweets, spammers can utilize some softwares such as AutoTwitter [33] and Twitter API [26] to automatically post more tweets on their profiles. Fig. 2(c) shows the distribution of the numbers of tweets of the *missed spammers* in each of three existing approaches, *all spammmers* and *all accounts*. From this figure, we can find that *missed spammers* (false negatives) post much more tweets than *all spammers*, even though the tweet numbers of *all spammmers* are much lower than that of *all accounts*. This observation also implies that spammers are trying to post more tweets to not to be recognized as spammers.

Mixing Normal Tweets: Based on observations of the missed spammers by the existing work, we can find that some of them post non-spam tweets to dilute their spam tweet percentage. Fig. 3(a) shows a real example of a spammer that posts famous quotes, "Winning isn't everything, but wanting to win is. – Vince Lombardi", between tweets containing links to phishing and

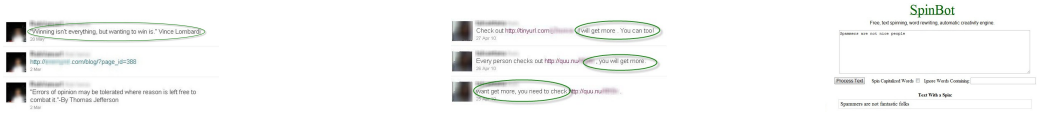


(a) Number of followers (b) Fofo Ratio (b) Number of tweets

Fig. 2. Profile-based Feature Examination on Three Existing Works

scam websites.

Posting Heterogeneous Tweets: In order to avoid detection features such as *tweet similarity* and *duplicate tweet count*, spammers use tools to 'spin' their tweets so that they can have heterogeneous tweets with the same semantic meaning but with totally different words. Fig. 3(b) shows a spammer that posts various messages encouraging users to sign up for a service that is eventually a trap to steal users' email addresses. Notice that the spammer uses three different phrases that have the same semantic meaning: "I will get more. You can too!", "you will get more.", and "want get more, you need to check". An example of automatical tools that can be used to create such heterogeneous tweets, called spin-bot, is shown in Fig. 3(c). By typing a phrase into the large text field and pressing "Process Text", a new phrase with the same semantic meaning and yet different words is generated in the small text field below.



(a) Mixing Normal Tweets (b) Posting Heterogeneous Tweets (c) Spin-bot

Fig. 3. Case Studies for Content-based Feature Evasion Tactics

This chapter showed statistical evidence that spammers are using tech-

niques to avoid detection. After analysis of the evasive spammers, hypotheses of their techniques were created. These hypotheses were then verified with data analysis. With these techniques in mind, the next chapter will design new features that make it more difficult for spammers to evade detection.

CHAPTER VI

DESIGNING A NEW DETECTION SCHEME

The previous chapters have analyzed existing detection schemes and identified tactics that spammers use to evade these detection schemes. With these tactics in mind, this chapter attempts to design a detection scheme that will be more robust against spammers' evasion techniques. The machine learning model is ideal because it can quickly be retrained against new adaptive spammers and automatically classify unknown accounts. The weakness of this model that the spammers are attacking is the feature set. The spammers change their behavior to make their features appear normal. In order to make a more robust detection scheme, more robust features are required. A robust feature should be difficult for the spammer to change or should be expensive for the spammer to change. A feature is difficult to evade if it requires a fundamental change in the way a spammer performs its malicious deeds. A feature is expensive to evade if evasion requires the spammer to spend money, time or resources in order to evade detection. The newly designed features include three Graph-based features, three Neighbor-based features, three Automation-based features and one Timing-based feature. The details of these features will be introduced in the following sections.

A. Graph-based Features

Twitter can be considered as a graph where each Twitter account i is a vertex and each follow relationship is a directed edge e. It is cheap and easy for a

spammer to change their tweeting behavior or add more followers, however, it is difficult for them to change their behavior in the graph. Based on this intuition, this work presents three graph-based features: local clustering coefficient, betweenness centrality and bi-directional links ratio.

1. Local Clustering Coefficient

The local clustering coefficient [34] of a vertex is the number of pairs of its neighbors that have edges between them to the number of pairs of neighbors that do not have edges between them. This is an intuitive feature because the neighbors of a legitimate account are more likely to know each than those of a spam account. For example, a Twitter user may follow all of his co-workers who also follow each other. This account will have a high local clustering coefficient. On the other hand, a spammer may follow random people that do not follow each other. This spammer will have a low local clustering coefficient.

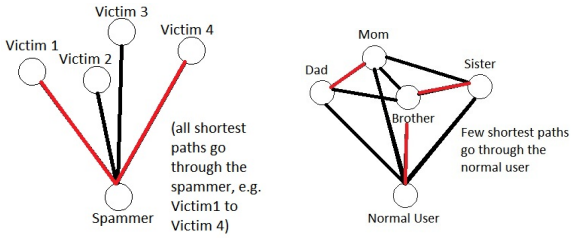
For each vertex v in the Twitter graph, its local clustering score can be computed by Equation (6.1), where K_v is the total degree of the vertex and $\{e_{ij}\}$ is the total number of the edges among all vertex v's neighbors.

$$LC(v) = \frac{2|\{e_{ij}\}|}{K_v \cdot (K_v - 1)} \tag{6.1}$$

2. Betweenness Centrality

Betweenness centrality [35] is a centrality measure of a vertex within a graph. Vertices that occur on many shortest paths between other vertices have a higher betweenness centrality than those that do not. This metric will reflect the position of the vertex in the graph. Nodes that occur in many shortest paths have

higher values of betweenness centrality. A Twitter spammer wil typically use a shotgun approach to finding victims, which means it will follow many other accounts without regard for whom they are or with whom these victims are connected. As a result, many of their victims are unrelated accounts, and thus their shortest path between each other is the average shortest path between all nodes in the graph. When the Twitter spammer follows these unrelated accounts, this creates a new shortest path of length 2 between any victim followee of the spam account and any other victim followee, through the spam account. This is illustrated in Figure 4 (a). Thus, the spam account will be on many such shortest paths between its neighbors and the betweenness centrality of the spammer will be high. On the other hand, a typical Twitter user may follow many people interested in the same topic and these people are more likely to have direct connections, which would not put the typical Twitter user on a shortest path between these users, giving the typical Twitter user a lower betweenness centrality than the typical spammer. This is illustrated in Figure 4 (b).



(a) Spammer (b) Normal

Fig. 4. Shortest Paths Between Neighbors of a Spammer (red) and a Normal User (green)

In a directed graph, betweenness centrality of each vertex v can be com-

puted by Equation 6.2, where δ_{st} is the number of shortest paths from s to t, and $\delta_{st}(v)$ is the number of shortest paths from s to t that pass through a vertex v, and n is the total number of vertexes in the graph.

$$BC(v) = \frac{1}{(n-1)(n-2)} \cdot \sum_{s \neq v \neq t \in V} \frac{\delta_{st}(v)}{\delta_{st}} \tag{6.2}$$

3. Bi-directional Links Ratio

As mentioned in Chapter IV, the number of bi-directional links has been used as a feature by existing work. However, Twitter spammers can easily evade this feature by purchasing followers and following them back. In order to make this feature a bit more robust, the number of bi-directional links is compared to the total number of users an account follows. The intuition behind this feature is that it is difficult and dangerous for Twitter spammers to increase their bi-directional links ratio. For each legitimate user they want to follow, they would have to gain another bi-directional link to keep this ratio the same. Thus they would need to purchase followers, incurring a cost, and then follow these purchased followers back. Following these purchased followers back also puts them in close relationship with likely low reputation or fake accounts, which is not good for their account's reputation. Thus, compared with the high number of users they follow, their bi-directional links ratio will be difficult to forge.

Bi-directional links ratio (R_{bilink}), can be computed in Equation 6.3.

$$R_{bilink} = \frac{N_{bilink}}{N_{fing}} \tag{6.3}$$

where N_{bilink} and N_fing denote the number of bi-directional links and the num-

ber of following.

B. Neighbor-based Features

The next set of features are based on the intuition that spammer can easily modify his own behavior, however he cannot easily modify the behavior of the accounts that he follows. These are called neighbor-based features and include: average neighbors' followers, average neighbors' tweets and followings to median neighbors' followers.

1. Neighbors' Follower Quality

Neighbors' follower quality, denoted as F_{qual}, of an account v represents the quality of an account's neighbors. Since an account's follower number usually reflects the account's popularity or quality, this feature can be quanitfied by the average number of followers of an account's neighbors. Legitimate accounts tend to follow accounts of high follower quality that have many followers unlike spammers who try to get the attention of normal users who have few followers. Thus, legitimate accounts typically have higher neighbors' follower quality than spammers.

Average neighbor's followers can be computed as Equation (6.4).

$$F_{qual}(v) = \frac{1}{|N_{fing}(v)|} \cdot \sum_{u \in N_{fing}(v)} N_{fer}(u) \tag{6.4}$$

where N_{fer} and N_fing denote the number of followers and followings, respectively.

2. Neighbors' Tweet Quality

Another good quality of a Twitter account is the number of Tweets. A legitimate user typically wants to follow users that will provide them with information, meaning they will Tweet often. On the other hand, spammers want to follow legitimate users who do not Tweet as much. For this reason, neighbors' tweet quality, denoted as T_{qual}, is the average number of Tweets of an accounts neighbors. Section VII shows that these two features can be evaded by following popular Twitter accounts, however, the spammer will need to buy more followers to keep his following follower ratio low.

3. Enhanced Neighbors' Follower Quality

Most of the users a spammer will follow will be potential victims, which are typical Twitter users. These users have small follower counts, thus the median number of followers of a spam account's followees, M_{nfer} will be small. However, some typical users may mostly follow their friends, other typical Twitter accounts and also have small values for M_{nfer}. A big difference between a typical Twitter account is the number of followees. Thus, the ratio between the number of followees and M_{nfer}, R_{fing_mnfer}, will be different for spammers and typical Twitter accounts. Typical Twitter accounts will follow few users compared to the large number of users that a typical spam account will follow, thus the value for R_{fing_mnfer} will be low for typical users and high for spammers. Other typical users may follow only popular Twitter accounts. Thus they will have a high value for M_{nfer}, but with their low number of followers, the value for R_{fing_mnfer} will remain low for this type of typical Twitter user as well.

This metric can be computed by Equation 6.5.

$$R_{fing\_mnfer} = \frac{N_{fing}}{M_{nfer}} \tag{6.5}$$

C. Automation-based Features

In order for a spammer to make a great profit, he must control many spam accounts at once. This is done through writing automated programs to control groups of accounts at once. These programs must use the Twitter API to make these accounts post tweets and follow users. The next features take advantage of this behavior pattern.

1. API Ratio

API ratio is the ratio of the number of tweets posted using API to the total number of tweets posted. As existing work [36] shows, many bots choose to use API to post tweets, so a high API ratio implies this account is more suspicious.

2. API URL Ratio

Some non malicious accounts may have created their own automated tools for posting harmless tweets to Twitter. Thus, the API tweets themselves must be analyzed to determine if they are spam. Thus, the ratio of API posts containing URLs is an indicator. A high API URL ratio indicates that the user is using API to automatically post spam.

3. API Tweet Similarity

Along the same logic, if tweets posted using API are more likely to be spam tweets, they should be analyzed more closely. The similarity between API tweets helps to determine if the API tweets are being automated. If they are similar, they are likely being automated by a spammer controlled program, other wise they may be coming from a legitimate user's program.

D. Timing-based Features

Similar to other timing-based features such as "tweet rate" presented in [13], we also design another feature named "following rate".

1. Following Rate

"Following rate" reflects the speed at which an account follows other accounts. An extremely high value of the following rate is suspicious since it indicates a user is following many other users in a short time and may be spamming. Twitter does not allow us to collect the times at which a user followed another user, thus the following rate must be estimated. To estimate the following rate, the number of followings of an account is divided by the age of the account (current time - time of creation).

These features will be a part of the feature set used for the new detection scheme. Next, the robustness of existing detection features and our newly designed features is formalized in Chapter VII. Then, existing effective features are combined with newly designed features to complete the new detection scheme which is then evaluated on the new dataset. The specific evaluation

results can be seen in Chapter VIII.

CHAPTER VII

ANALYSIS OF DETECTION SCHEMES

In this chapter, the existing detection schemes and the newly designed detection scheme are analyzed. Since each scheme is based on the features that they use to describe Twitter accounts, the features themselves will be analyzed. The robustness of each feature is formalized and this chapter shows that the new features are more robust than the older features.

A. Formalizing Feature Robustness

In this section, a formal definition of a robust detection feature is given and each feature is evaluated based on this definition.

1. Formalizing Robustness

Before analyzing the robustness of each feature in detail, a model is created to quantify the robustness of a detection feature. A robust feature should either be difficult for the spammer to change or it should be expensive to change. Spammers are constantly trying to avoid detection while at the same time trying to achieve malicious goals. Based on these priorities, the robustness of each feature F, denoted as $R(F)$, can be viewed as the tradeoff between a spammer's cost $C(F)$ to avoid detection and the profits $P(F)$ earned by achieving malicious goals. Thus, the robustness of each feature can be computed by Equation 7.1. This is intuitive because the higher the cost is compared to the profit, the more

robust the feature.

$$R(F) = C(F) - P(F) \tag{7.1}$$

Thus, if the cost of evading a detection feature is higher than the profits, then the feature is robust. To quantify a successful evasion of a particular detection feature F, a threshold value T_F is used to denote the value a spammer needs to obtain to evade the feature. The cost for a spammer to evade detection includes three types of costs: monetary cost, operational cost and time. The monetary cost is mostly from having to pay to obtain high numbers of followers. The operational cost is related to posting tweets or following specific accounts. Let C_{twt} and C_{follow} denote the cost for a spammer to post one tweet or follow one specific account and let C_{fer} denote the cost for a spammer to obtain one follower. A spammer's profits are achieved by attracting the attention of legitimate accounts, thus, a Twitter spammer's profits can be measured by the number of users they can follow and the number of spam tweets that they can post. P_{fing} and P_{mt} are used to denote the profit of supporting one following account and posting one spam tweet, respectively. Let N_{fing} and N_{mt} denote the number of accounts that the spammer desires to follow and the number of malicious tweets that the spammer desires to post.

This section analyzes the robustness of each of the following six categories of features: profile-based features, content-based features, neighbor-based features, graph-based features, timing-based features and automation-based features. The summary of these features can be seen in the table on page 49. In this table, features labeled with "this" means this feature is included in this work's detection scheme.

2. Profile-based Features

As described in Chapter V, the spammer usually evades this type of detection feature by obtaining more followers. The following follower ratio of an account is a representative feature of profile-based features. According to Equation 7.1, the robustness of the following follower ratio detection feature(F_3) can be computed by Equation (7.2).

$$R(F_3) = \frac{N_{fing}}{T_{F_3}} \cdot C_{fer} - N_{fing} \cdot P_{fing} \quad (T_{F_3} \geq 1) \tag{7.2}$$

For example, if the maximum threshold for F_3, T_{F_3} is 2, then for every 2 potential victims the spammer follows, the spammer will have to buy 1 follower. Thus, $\frac{N_{fing}}{T_{F_3}}$ is the number of followers that the spammer will have to purchase. Table II, shows that C_{fer} is inexpensive. The number of users that must be exposed to the spam content before one is tricked into becoming a victim has many factors and is unknown. However, even if only 1 user in every 1,000 viewers becomes a victim, then the spammer would need to follow 1,000 users which means he would need to buy 500 followers(assuming $T_{F_3} = 2$) which costs \$23.50 per victim. This amount is based on the most expensive price per follower from Table II in Chapter V. This cost is much smaller than the 1,000s of dollars that could be stolen from the single victim through identity theft [9]. This shows that this feature is not robust because the cost of evasion is much smaller than the potential profits. Similar conclusions can be drawn for features F_1, F_2 and F_4.

For the feature F_6, since the age of an account is determined by the time when the account is created, which cannot be changed or modified by the spam-

mers, this feature is difficult to evade. A spammer may try to evade this feature by obtaining an existing Twitter account. However, unlike obtaining followers, obtaining a specific Twitter account will be very expensive. For example, the bid value to purchase a Twitter account that steadily has over 1,390 followers is $1,550 [37]. Another way to evade account age is to create a set of accounts and wait for several months before using these accounts. However, this costs the spammer time, which makes the spammer less effective. Thus the account age feature is fairly robust.

3. Content-based Features

Content-based features can be divided into two types: signature-based features (F_7, F_8, F_9, and F_{10}) and similarity-based features (F_{11}, and F_{12}). As discussed in Chapter V, both types of features can be evaded by automatically posting signature avoidng tweets or diverse tweets. Also, by using these tactics, the spammers post more tweets, thus evading the feature of the number of tweets (F_5).

Without loss of generality, the analysis of the robustness of the URL_ratio feature (F_7) will be used to demonstrate the analysis of this type of feature. In order to maintain a low URL_ratio, one must post non-spam tweets that do not contain URLs. However, the more non-spam tweets one posts, the less likely a potential victim is to see the spam tweets. If a spammer posts N_{st} spam tweets containing malicious URLs and N_{twt} total tweets, then the robustness of the URL_ratio can be computed by Equation 7.3.

$$R(F_7) = \frac{N_{mt}}{T_{F_7}} \cdot C_{twt} - \frac{N_{mt}}{N_{twt}} \cdot P_{mt} \quad (T_{F_7} \leq 1) \tag{7.3}$$

The cost to evade this feature is small. To evade this feature, a spammer needs only to post a sufficient number of non-spam tweets which is inexpensive and easy to do. However, the power of this feature comes from forcing the spammer to dilute his spam tweets with non-spam tweets, which decreases the overall profit that an account will make. This is shown in Equation 7.3 because when a spammer posts many non-spam tweets, the value of $\frac{N_{mt}}{N_{twt}}$ decreases and thus the profit decreases. Therefore, the URL_ratio feature is fairly robust.

4. Graph-based Features

The graph-based features can be divided into two types: reciprocity-based features (bi-directional link count and bi-directional link ratio) and position-based features (F_{15} and F_{16}). First the robustness of reciprocity-based features will be dicussed.

Let C_{BiLink} denote the cost to obtain one bi-directional link. The robustness of bi-directional link count (F_{13}) can then be computed in Equation 7.4.

$$R(F_{13}) = T_{F_{13}} \cdot C_{BiLink} - (N_{fing} - N_{bi-link}) \cdot P_{fing} \tag{7.4}$$

This equation shows that the cost of evading the bi-directional link count is C_{BiLink} times $T_{F_{13}}$, the number of bi-directional links needed to evade this feature. The profit is affected by the number of bi-directional links required. Each bi-directional link forces a spammer to follow a social butterfly or a created dummy account. This spammer could have instead followed another possible

victim. Thus this feature is only slightly robust.

The next feature, bi-directional link ratio (F_{14}), is an improved version of bi-directional link count. The robustness of the bi-directional link ratio feature is calculated in Equation 7.5.

$$R(F_{14}) = T_{F_{14}} \cdot N_{fing} \cdot C_{BiLink} - (N_{fing} - N_{bi-link}) \cdot P_{fing} \tag{7.5}$$

In this equation, the profit is the same as it is for bi-directional link count, however, the cost is different. Using a ratio instead of just the count requires that the number of bi-directional links needed scales with the number of users an account follows. Thus, for a spam account that follows many accounts, he will need to obtain many bi-directional links to appear normal. The average value of the bi-directional links ratio is 22.1% [18] and spammers usually follow a large number of accounts, thus, the spammers need to obtain many more bi-directional links to show a normal bi-directional links ratio. This feature can still be evaded by gaining enough bi-directional links through following social butterflies and creating dummy accounts. However, the spammer will have to dedicate approximately 20% of the users he follows to these accounts that will follow him back. That 20% of accounts could have been potential victims, which makes this spammer less effective.

Next the position-based features will be evaluated. These features include betweenness centrality (F_{15}) and clustering coefficent (F_{16}). The strength behind these features lyes in their abstractness and difficulty to change. The average Twitter user does not concern themselves with their position in the graph, which affects these values. Also, there is no benefit for a spammer to

change these values either, unless they are aware that these indicators are being used to detect their accounts. Suppose a spammer was aware of these features and wanted to avoid them. In order to have a smaller betweenness centrality or a higher clustering coefficient, a spammer would have to ensure that he followed accounts that were related to other accounts that he follows. This is possible, however, a spammer cannot be as effective if the users that he can follow are limited. Thus, these features are robust.

5. Neighbor-based Features

The first two neighbor-based features reflect the quality of an account's friends choice and were discussed in Section VI. Let N_{follow} denote the number of high quality accounts (accounts with many followers) that a spammer needs to follow to get a high enough A_{nfer} to evade feature F_{17}, then the robustness of F_{17} can be computed as Eq.(7.6).

$$R(F_{17}) = N_{follow} \cdot C_{follow} \tag{7.6}$$

Since there are many popular accounts with many followers, N_{follow} and C_{follow} could be small. Thus, as long as the spammers know about this detection feature, they can evade it easily. Similar results can be gained for feature F_{18}.

However, for feature F_{19}, since the median is used instead of the mean of the neighbors' follower count, at least half of the users they follow must be popular accounts to evade this feature. Since spammers follow many users, the cost of evading this feature will be very high and the profit will decrease dramatically for the spammers to evade this feature. So, feature F_{19} is fairly

robust.

6. Timing-based Features

The timing-based features are related to spammers' update behavior. In order to evade the following rate feature (F_{20}), a spammer will need to follow users at a slow pace and in order to evade the tweet-rate feature (F_{21}) a spammer can simply post tweets at a normal rate as well. However easy these features are to evade, they require the spammer to slow down his operation in order to do so, which costs the spammer valuable time. For these reasons, feature F_{20} and F_{21} are both relatively robust.

7. Automation-based Features

In order to publish the amount of tweets required for a successful spam campaign, many spammers use automated programs to manage several spam accounts. These types of software are also helpful in evading content-based features. In order to create custom automation programs, spammers will use the Twitter API as it is the best way to interface with Twitter programmatically.

Let C_{twt_web} and C_{twt_api} denote the cost of using web and api to post one tweet. Since it takes time to manually log in to Twitter for each account and post a tweet $C_{twt\_web} \gg C_{twt\_api}$. If a spammer wants to use API to post spam or malicious tweets on N_{spam} different spam accounts, which is very common in practice, then the robustness of feature F_{22} can be computed as Equation 7.7.

$$R(F_{22}) = N_{spam} \cdot [\frac{N_{mt}}{T_{F_{22}}} \cdot (1 - T_{F_{22}}) \cdot C_{twt\_web} + N_{mt} \cdot C_{twt\_api}] - N_{spam} \cdot N_{mt} \cdot P_{mt} \quad (7.7)$$

Since fewer legitimate accounts would use the Twitter API to post tweets, the value of $T_{F_{22}}$ should be small. In this way, since $C_{twt\_web} \gg C_{twt\_api}$ if a spammer wants to control many spam accounts and post a large number of tweets, the cost will be relatively high. The conclusions for the rest of this type of feature are similar.

It is noticeable that only using feature F_{22} would lead to false positives since legitimate accounts may also use API to post tweets. However, by combining features F_{22}, F_{23}, and F_{24}, it will decrease those false positives since only a few legitimate accounts would use the Twitter API to post very similar tweets containing URLs as spammers do.

Using the same method as above, the robustness of all features has been categorized into the following three scales: low, medium and high. The summary of this information can be seen in Chapter VIII. The information provided in that chapter shows that several of the features used by existing works are not very robust, such as the number of tweets and the number of bi-directional links and several of the new features presented in this work are robust such as clustering coefficient and bi-directional link ratio.

Table III. Detection Feature Robustness

Index	Category	Feature	Work	Robustness
F_1	Profile	the number of followers (N_{fer},)	[14]	Low
F_2	Profile	the number of followings (N_{fing})	[15], [14]	Low
F_3	Profile	fofo ratio (R_{fofo})	[12], [15], this	Medium
F_4	Profile	reputation (Rep)	[14]	Medium
F_5	Profile	the number of tweets (N_{twt})	[15], this	Low
F_6	Profile	age	[12], this	High
F_7	Content	URL ratio (R_{URL})	[12], [15], [14], this	Low
F_8	Content	unique URL ratio	[12], this	Low
F_9	Content	hashtag(#) ratio	[14]	Low
F_{10}	Content	reply(@) ratio, [14]	[12]	Low
F_{11}	Content	tweet similarity (T_{sim})	[12], [15], this	Low
F_{12}	Content	duplicate tweet count	[14]	Low
F_{13}	Graph	the number of bi-directional links (N_{bilink})	[12]	Low
F_{14}	Graph	bi-directional links ratio (R_{bilink})	this	Medium
F_{15}	Graph	betweenness centrality (BC)	this	High
F_{16}	Graph	clustering coefficient (CC)	this	High
F_{17}	Neighbor	average neighbors' followers (A_{nfer})	this	Low
F_{18}	Neighbor	average neighbors' tweets (A_{ntwt})	this	Low
F_{19}	Neighbor	(R_{fing_mnfer})	this	High
F_{20}	Timing	following rate (FR)	this	Low
F_{21}	Timing	tweet rate (TR)	[12], this	Low
F_{22}	Automation	API ratio (R_{API})	this	Medium
F_{23}	Automation	API URL ratio (R_{API_URL})	this	Medium
F_{24}	Automation	API Tweet Similarity (T_{api_sim})	this	Medium

CHAPTER VIII

EVALUATION

This chapter, presents an evaluation of the performance of the new detection scheme. This scheme uses a feature set that contains 9 existing effective features and 10 newly designed features including the following features F_2 F_8, F_{11} and F_{14} F_{24}, which were chosen using feature selection techniques. The names of the features can be seen in Table III.

The features set is evaluated by running the machine learning technique on two different data sets: Data set I and Data set II. Data set I refers to the 5,500 Twitter accounts that are described in Chapter IV. To decrease the effects of sampling bias and show the quality of the new detection feature schema without using URL analysis as ground truth, another dataset containing 35,000 Twitter accounts was crawled and 3,500 accounts were randomly selected to build another data set, denoted as Data set II.

A. Comparison of Detection Schemes

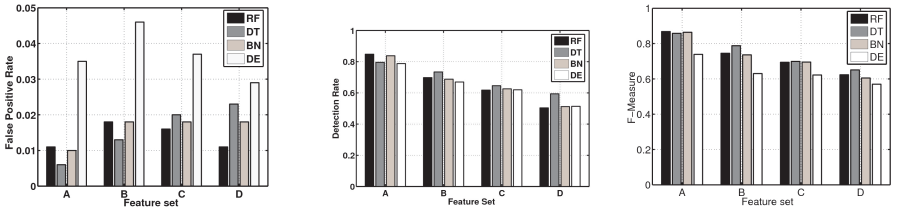
In this section, three experiments are performed using Dataset I to evaluate the new detection scheme: performance comparison, feature effectiveness and learning curve.

1. Performance Comparison

In this experiment, the new detection scheme is compared against the existing schemes: [12] using 10 features, [15] using 8 features and [14] using 7 features.

The details of the features used in these three works can be seen in Table III. The evaluation is performed using four different machine learning algorithms: *Random Forest (RF)*, *Decision Tree (DT)* , *Bayes Net (BN)* and *Decorate (DE)*. In the results, the new scheme will be labeled as A, [12] will be labeled as B, [15] will be labeled as C and [14] will be labeled as D.

For each machine learning classifier, *ten-fold cross validation* is used to find the *false negatives*, *false positives* and *F-measure*. As seen in the Figure 5, the new approach outperforms the existing schemes.



(a) False Positive Rate (b) Detection Rate (c) F-Measure

Fig. 5. Performance Comparison with the Existing Approaches

While maintaining a low false positive rate, the detection rate of the new scheme increases to 85%, compared with the detection rate of 51% for the worst detector (D [14]) and the detection rate of 73% for the best other existing detector (B [12]). Here detection rate refers to the percentage of spammers that were correctly classified. F-measure is a metric that balances precision, the percentage of users that the scheme classified as spammers that actually are spam with recall, the percentage of spammers that were correctly identified. The F-measure used here is F-1 which balances these values equally. The new approach has the highest F-1 rank with each algorithm. This better performance comes from the ability of the new detection scheme to be robust against evasion

tactics that are able to avoid the existing detection schemes. Next the individual features are evaluated using feature selection techniques.

2. Evasive Users Caught

Figure 6 shows the number of spammers that evaded each feature set. This figure also shows how many of those evasive spammers were caught by the new detection scheme. The new algorithm is able to catch a majority of the spammers that the existing detection schemes missed. Notice that the difference between the number of evasive spammers and the number of those spammers caught is similar for each existing scheme. This shows that there is a group of super evasive spammers that evaded each scheme including the new scheme. Detecting these super evasive spammers is left for future work.

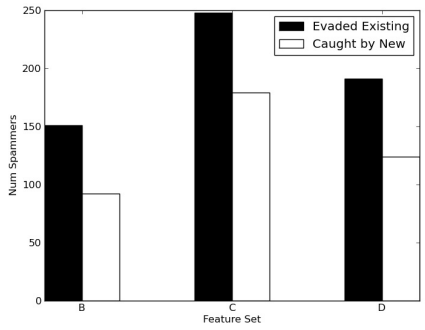


Fig. 6. Evasive Spammers Caught

3. Feature Evaluation

To further validate the effectiveness of the newly designed features, the performance of two feature sets is compared. The first feature set consists of the

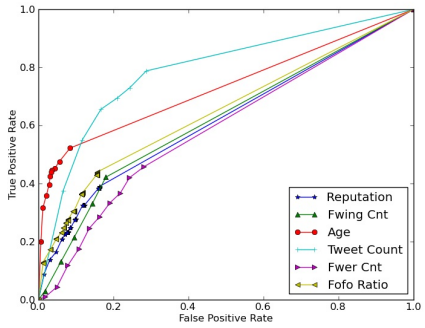
features used in the previous experiment without the newly designed features. The second feature set consists of all features used in the previous experiment. Table IV, shows that for each classifier, with the addition of the newly designed features, the detection rate increases more than 10%, while maintaining an even lower false positive rate. This observation implies that the improvement of the detection performance is indeed proportional to our newly designed features rather than the combination of several existing features.

Table IV. Comparison Without and With New Features

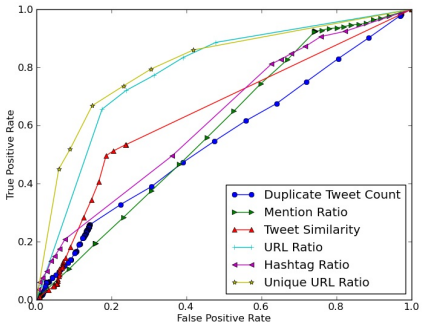
	Without Our Features			With Our Features		
Algorithm	FPR	Detection Rate	F-Score	FPR	Detection Rate	F-Score
Decorate	0.017	0.738	0.774	0.010	0.858	0.877
Random Forest	0.012	0.728	0.786	0.006	0.836	0.884
Decision Tree	0.015	0.702	0.757	0.011	0.846	0.866
BayesNet	0.040	0.356	0.730	0.023	0.784	0.777

Next, each of the features are evaluated individually using the *Receiver Operating Characteristic* (ROC) curve. This experiment uses a decision stump based on 1 feature and varies the threshold for classification. Figure 7 shows a graph for each feature type containing the ROC curves for each feature. Notice that the best performing features are the profile features and the content features. The neighbor features perform the worst. However, the purpose of the newly designed features is not to detect spammers better in general but to detect evasive spammers.

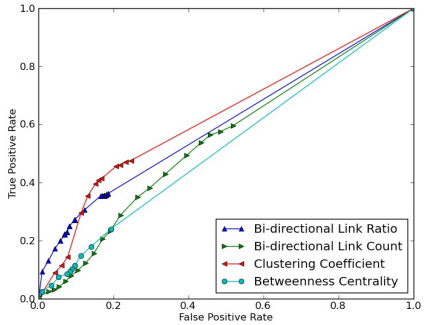
The next experiment shows how effectively the features distinguish evasive spammers from normal users. To do this a data set consisting of spammers that



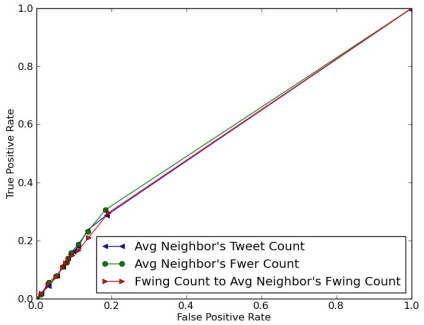
(a) Profile Features



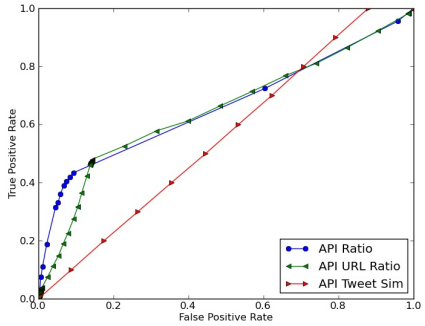
(b) Content Features



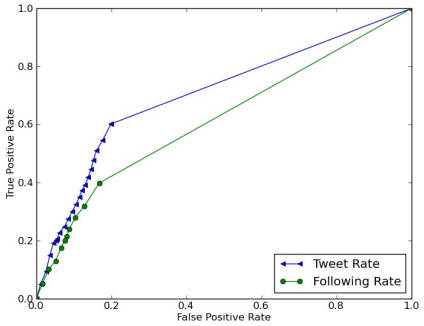
(c) Graph Features



(d) Neighbor Features



(e) Automation Features



(f) Timing Features

Fig. 7. Individual Feature ROC Curves

Table V. Feature Rank

Feature	Chi Square	Info Gain	AUC
number of followings	13	10	8
fofo ratio	8	9	15
number of tweets	7	7	10
account age	9	8	7
URL ratio	5	3	2
unique URL ratio	6	5	1
tweet similarity	11	12	9
bi-directional links ratio	12	15	12
betweenness centrality	18	18	17
clustering coefficient	10	14	6
average neighbors' followers	16	16	13
average neighbors' tweets	3	4	5
followings to median neighbors' followers	17	17	16
following rate	14	11	18
tweet rate	4	6	11
API ratio	1	1	3
API URL ratio	2	2	4
API tweet similarity	15	13	14

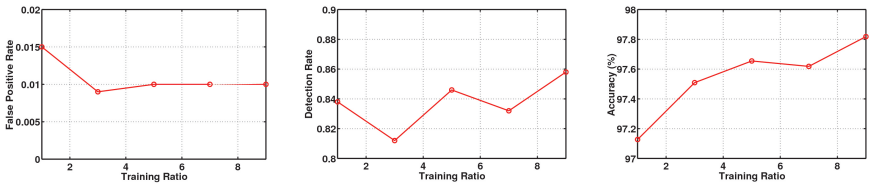
were misclassified by at least one of the previous algorithms and normal users is used. Table V shows the rank of each of the features based on three feature selection techniques: Chi Square, Information Gain and area under the ROC curve (AUC). Notice that while the neighbor-based features did not perform well in the previous experiment, average neighbors' tweets is ranked highly according to each feature selection technique. Also, while the best features in the previous experiment were profile-based and content-based features, here the best features are automation-based features. This shows that the new features are able to classify the evasive spammers better than the features used in previous works.

4. Learning Curve

This section presents an experiment to show the steadiness of the new detection scheme with varying amounts of training data. In order to show this, the ratio of training data to testing data is varied and classifcation results are obtained. Figure 8 shows the results. The detection rate increases slightly with more training data, however, even with a small amount of training data it never drops below 80%, still better than the best previous work. Accuracy increases more drastically, indicating that more normal users are being misclassified. Also, the false positive rate never climbs above 2%, even with equal amounts of training and testing data.

B. Real World Evaluation

In this section, the new detection scheme is evaluted on a separate data set containing 3,500 unclassified Twitter accounts. This experiment is a real world



(a) False Positive (b) Detection Rate (c) Accuracy

Fig. 8. Effect of Varying the Training Ratio

evaluation of the new detection scheme, since no accounts have been pre-labeled. The classifier is first trained using Dataset I, then the unclassified instances in Dataset II are classified using the BayesNet classifier. Those users labeled as spammers are then manually analyzed to determine the quality of the results. The results of this analysis are shown in Table VI.

The classifier identified 70 accounts as spam accounts. After analyzing the URLs that these accounts posted, GSB found 17 of them posted malicious URLs in their tweets. This does not mean that the rest are innocent users, they simply are using URLs that have yet to be identified by GSB. Through manual analysis over half of reported spammers were actually verified as spam accounts. However, the rest of the accounts were not ordinary Twitter accounts. In fact, 25 of the users were identified as advertisers. These are legitimate accounts

Table VI. Classifier Effectiveness

Total Spammer Predictions	70
Verified as Spammers	37
Promotional Advertisers	25
Identified by GSB	17
Benign	8

that are advertising a legitimate business on Twitter. Many advertisers behave similarly to spammers, which makes it difficult for an automatic classifier to distinguish between the two.

These results show that the new detection scheme has a high Bayesian detection rate at 88.6% (62/70). The 8 benign Twitter accounts that were labeled as spam were investigated to determine the reason for their misclassification. This analysis shows that all of them have odd behavior, typical of Twitter spammers, but do not appear to have malicious intentions. Specifically, 6 of these are actively tweeting about a particular topic (i.e. skateboarding) and thus have similar tweets and post URLs often. The other 2 have posted very few tweets, yet have a large number of followers and followees with a high following follower ratio. The number of accounts classified as normal is too large for manual investigation to determine false negative rate. However, in these types of spam detection systems, the important factor is the false positive rate. This is because it is better to not suspend a spammer than it is to suspend a normal user since being wrongly suspended may irritate a normal user more than spam. For this reason, the analysis focuses on those users that were identified as spam instead of those identified as normal.

In practical situations, a major concern is the length of time it takes to classify users because there are many users in an actual OSN. This is the reason that the machine learning technique is popular, because most machine learning classifiers are able to classify instances quickly, even if it takes more time to train the classifier. To confirm this the training and classifying time for data set II was recorded. The training time was 10.81s and the classification took so

little time the Unix *time* function recorded 0.00s. The relatively long training time is not a concern because the training time does not scale with the number of instances to classify.

This chapter showed that the new detection scheme was able to out perform the existing schemes in all metrics. Also, the new detection scheme proved effective at identifying spammers in an unclassified data set. Also, the tribulations of detecting spam in an environment where advertising is prevalent were discussed.

CHAPTER IX

LIMITATION AND FUTURE WORK

Due to the practical limitations, the dataset includes only a portion of the data on Twitter. Also, despite attempts to reduce the sampling bias and the use of two separate data sets, the data may still be slightly biased. However, collecting an ideal large data set from a real, dynamic OSN such as Twitter with neither any errors nor any bias is difficult if not impossible.

As described in Chapter IV, it is also challenging to achieve comprehensive ground truth for the Twitter spammers. The collected spammers belong to one major type of spammers so that the number of collected Twitter spammers is a lower bound of the number of real spammers in the data set. However, even for a subset of real spammers, this work shows that they have utilized different tactics to evade existing detection techniques. Also, the evaluation results on these spammers have shown that the newly designed features can be used as an effective supplement to existing detection features to detect evasive Twitter spammers.

While graph-based features such as local clustering coefficient and betweenness centrality are relatively difficult to evade, these features are also expensive to extract. Thus, a sampling technique is used to calculate these metrics, which may decrease the accuracy of the values of these features. Also, since the time when an account follows another one is not available, an approximation is used to calculate the following rate. For one thing, even this feature may be not perfectly accurate, but an approximate value of this feature can still reflect how radically an account is trying to increase its following number.

Despite these limitations this work presents the first analysis of the evasion tactics being used in OSNs. This work lays a foundation for future works to discover and investigate new evasion tactics as they reveal themselves. For future work, to overcome these limitations, better crawling strategies must be designed and larger, more comprehensive data sets must be used. Also, as spammers continue to evolve their evasion tactics, more robust features will need to be developed. Since this work focuses on malicious spammers, the evasion tactics of other types of spammers should be studied.

CHAPTER X

CONCLUSION

In this paper, new robust features are designed to detect evasive Twitter spammers through an in-depth analysis of the evasion tactics utilized by current Twitter spammers. This work provides the first detailed analysis on how spammers can evade existing detection features. Also, the detection rate of three state-of-the-art solutions are examined which show that some Twitter spammers have indeed evolved to evade detection. Then, according to that analysis, several new features are designed and the effectiveness of these features is shown through the evaluation experiments. Also, a novel formalization of the robustness of a detection feature is designed and all features used by this work and existing works are measured agains that formalization.

This work shows that while there is a lot of work being done on spam detection in OSNs, the spammers are also working hard to avoid this detection. As the arms race continues, more works like this will be required to keep up with the evolving OSN spammers. In order to acheive effective results, researchers and OSNs need to work together to design OSN features that make it easier to distinguish a spammer from a normal user.

REFERENCES

[1] Jeffrey Heer and Danah Boyd. Vizter: Visualizing online social networks. *IEEE Information Visualization*, 2005.

[2] Business Insider. Facebook Q1 Profits Down, Operating Income Down. http://www.businessinsider.com/live-facebook-q1-revenues-profits-down-2012-4. Accessed: 04/2012.

[3] National Public Radio. Forgein Policy: Scramble to Silence Cairo Protests. http://www.npr.org/2011/01/28/133306415/foreign-policy-scramble-to-silence-cairo-protests. Accessed: 06/2012.

[4] John D. Sutter. A New Look at Spam by the Numbers. http://scitech.blogs.cnn.com/2010/03/26/a-new-look-at-spam-by-the-numbers/. Accessed: 06/2012.

[5] Ryan Singel. Botnet Over Twitter. http://www.wired.com/threatlevel/2009/08/botnet-tweets/. Accessed: 06/2012.

[6] Graham Cluley. Acai Berry Spammers Hack Twitter Accounts to Spread Adverts. http://nakedsecurity.sophos.com/2009/05/24/acai-berry-spammers-hack-twitter-accounts-spread-adverts/. Accessed: 06/2012.

[7] Gregg Keizer. Koobface Worm to Users: Be My Facebook Friend. http://www.computerworld.com/s/article/9128842/Koobface_worm_to_users_Be_my_Facebook_friend. Accessed: 06/2012.

[8] Malware Survival. New Koobface Injects into All Browser Searches. http://malwaresurvival.net/2011/01/21/new-koobface-injects-into-all-browser-searches/. Accessed: 06/2012.

[9] Robert Hammond. *Identity Theft: How to Protect Your Most Valuable Asset*. Career Press, 2003.

[10] Twitter. The Twitter Rules. http://support.twitter.com/entries/18311-the-twitter-rules. Accessed: 06/2012.

[11] Meeyoung Cha, Hamed Haddadi, Fabricio Benevenuto, and Krishna P. Gummadi. Measuring User Influence in Twitter: The Million Follower Fallacy. *Int'l AAAI Conference on Weblogs and Social Media (ICWSM)*, 2010.

[12] K. Lee, J. Caverlee, and S. Webb. Uncovering Social Spammers: Social Honeypots + Machine Learning. *ACM SIGIR Conference (SIGIR)*, 2010.

[13] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. Detecting Spammers on Twitter. *Collaboration, Electronic Messaging, Anti-Abuse and Spam Conffference (CEAS)*, 2010.

[14] A. Wang. Don't follow me: Spam Detecting in Twitter. *Int'l Conferene on Security and Cryptography (SECRYPT)*, 2010.

[15] G. Stringhini, S. Barbara, C. Kruegel, and G. Vigna. Detecting Spammers On Social Networks. *Annual Computer Security Applications Conference (ACSAC'10)*, 2010.

[16] Buyafollower.com. Buy a Follower. http://buyafollower.com/. Accessed: 06/2012.

[17] SpinBot Blog. Tweet Spinning Your Way to the Top. http://blog.spinbot.com/2011/03/tweet-spinning-your-way-to-the-top/. Accessed: 06/2012.

[18] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a Social Network or a News Media? *Int'l World Wide Web (WWW '10)*, 2010.

[19] Twitter. How To Report Spam On Twitter. http://support.twitter.com/entries/64986. Accessed: 06/2012.

[20] Blocky. Blocky. http://blocky.elliottkember.com/. Accessed: 10/2010.

[21] G. Koutrika, F. Effendi, Z. Gyongyi, P. Heymann, and H. Garcia-Molina. Combating Spam in Tagging Systems. *Int'l Workshop on Adversarial Information Retrieval on the Web (AIRWeb'07)*, 2007.

[22] Fabricio Benevenuto, Tiago Rodrigues, Virgflio Almeida, Jussara Almeida, Chao Zhang, and Keith Ross. Identifying Video Spammers in Online Social Networks. *Int'l Workshop on Adversarial Information Retrieval on the Web (AirWeb'08)*, 2008.

[23] Fabrcio Benevenuto, Tiago Rodrigues, Virgflio Almeida, Jussara Almeida, and Marcos Gonalves. Detecting Spammers and Content Promoters in Online Video Social Networks. *ACM SIGIR Conference (SIGIR)*, 2009.

[24] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Zhao. Detecting and Characterizing Social Spam Campaigns. *Proceedings of ACM SIGCOMM IMC (IMC'10)*, 2010.

[25] Erick Schonfeld. Costolo: Twitter Now Has 190 Million Users Tweeting 65 Million Times a Day. http://techcrunch.com/2010/06/08/twitter-190-million-users/. Accessed: 06/2012.

[26] Twitter. Twitter API. https://dev.twitter.com/. Accessed: 06/2012.

[27] J. Leskovec and C. Faloutsos. Sampling from Large Graphs. *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2006.

[28] Twitter. Twitter Public Timeline. http://twitter.com/public_timeline. Accessed: 06/2012.

[29] Google Developers. Google Safe Browsing API. https://developers.google.com/safe-browsing/. Accessed: 06/2012.

[30] PhishTank. PhishTank. http://www.phishtank.com/. Accessed: 06/2012.

[31] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.

[32] Twitter. The 2000 Following Limit Policy On Twitter. http://support.twitter.com/groups/32-something-s-not-working/topics/117-following-problems/articles/66885-follow-limits-i-can-t-follow-people. Accessed: 06/2012.

[33] AutoTwitter. Auto Twitter. http://www.autotweeter.in/. Accessed: 06/2012.

[34] Steven Strogatz D. J. Watts. Collective dynamics of 'small-world' networks. *Nature*, pages 440–442, 1998.

[35] M.E.J. Newman. *Networks: An Introduction*. UK: Oxford University Press, 2010.

[36] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia. Who is Tweeting on Twitter: Human, Bot, or Cyborg? *Annual Computer Security Applications Conference (ACSAC'10)*, 2010.

[37] PotPieGirl. Twitter Account For Sale. http://www.potpiegirl.com/2008/04/buy-sell-twitter-account/. Accessed: 06/2012.

VITA

Robert Chandler Harkreader received his Bachelor of Science degree in Computer Engineering from Texas A&M University at College Station in 2009. He has received his Master of Science degree in Computer Science also at Texas A&M University.

Mr. Harkreader may be reached at 301 Harvey R. Bright Building, College Station, TX 77843-3112. His email address is bharkreader@gmail.com.