Die Free or Live Hard? Empirical Evaluation and New Design for Fighting Evolving Twitter Spammers

Chao Yang, Robert Chandler Harkreader, Guofei Gu

Success Lab, Texas A&M University
{yangchao, bharkreader, guofei}@cse.tamu.edu

Abstract. To date, as one of the most popular Online Social Networks (OSNs), Twitter is paying its dues as more and more spammers set their sights on this microblogging site. The Twitter spammers can achieve their malicious goals such as sending spam, spreading malware, hosting botnet command and control (C&C) channels, and launching other underground illicit activities. Due to the significance and indispensability of detecting and suspending those spam accounts, many researchers along with the engineers in Twitter Corporation have devoted themselves to keeping Twitter as spam-free online communities. Most of the existing studies utilize machine learning techniques to detect Twitter spammers. "While the priest climbs a post, the devil climbs ten". Twitter spammers are evolving to evade existing detection features. In this paper, we first make a comprehensive and empirical analysis of the evasion tactics utilized by Twitter spammers. We further design several new detection features to detect more Twitter spammers. In addition, to deeply understand the effectiveness and difficulties of using machine learning features to detect spammers, we analzye the robustness of 24 detection features that are commonly utilized in the literature as well as our proposed ones. Through our experiments, we show that our new designed features are much more effective to be used to detect (even evasive) Twitter spammers. According to our evaluation, while keeping an even lower false positive rate, the detection rate using our new feature set significantly increases to 85%, compared with a detection rate of 51% and 73% for the worst existing detector and the best existing detector, respectively. To the best of our knowledge, this work is the first empirical study and evaluation of the effect of evasion tactics utilized by Twitter spammers and is a valuable supplement to this line of research.

1 Introduction

Online social networking websites (OSNs), such as Twitter, Facebook and LinkedIn, are now part of many people's daily routine: from posting their recent experiences, finding out what friends are up to and keeping track of the hottest trends, to viewing interesting photos or videos. Twitter, a microblogging service founded in 2006, is one of the most popular and fastest growing online social networks, with more than 190 million Twitter accounts, tweeting 65 million times a day [8].

Twitter spammers have utilized OSNs as the new platform to achieve their malicious goals such as sending spam [2], spreading malware [13], hosting botnet command and control (C&C) channels [5], and performing other illicit activities [32]. All these malicious behaviors may cause significant economic loss to our society and even threaten national security. In August of 2009, nearly 11 percent of all Twitter posts were spam [1]. In May of 2009, many innocent users' accounts on Twitter were hacked to spread advertisements [2]. In February of 2010, thousands of Twitter users, such as the Press Complaints Commission, the BBC correspondent Nick Higham and the Guardian's head of audio Matt Wells, have seen their accounts hijacked after a viral phishing attack [22].

Many researchers along with engineers from Twitter Corporation have devoted themselves to keep Twitter as a spam-free online community. Their efforts have attempted to protect legitimate users from useless advertisements, pornographic messages or links to phishing or malicious websites. Twitter has published their definitions of spam accounts and The Twitter Rules [16], and also adopts a series of methods including Spammer Report [14] to protect its users from spam and abuse. Any account engaging in the abnormal activities specified by Twitter are subject to temporary or even permanent suspension. Meanwhile, according to Twitter spammers' special behavior, many existing research studies, such as [28, 35, 25, 38, 37], also utilize machine learning techniques to detect Twitter spammers by designing corresponding detection features.

"While the priest climbs a post, the devil climbs ten." This proverb illustrates the struggle between security researchers and their adversaries – spammers in this case. The arms race nature between the attacks and defenses encourages Twitter spammers to work hard to evade detection. In fact, many Twitter spammers are evolving to utilize new tactics or tools to evade existing detection features. For example, Twitter spammers can evade the detection feature of "the ratio of the number of followings to the number of followers" by purchasing followers [6] or exchanging followers [21]. Also, Twitter spammers can evade the detection feature of "the number of duplicate tweets" by using tools like Spinbot [17] to automatically post tweets with the same meaning but different words. These evasion tactics motivate us to design new, more robust features to detect Twitter spammers.

In this paper, we plan to design more robust detection features to detect evasive Twitter spammers through in-depth analysis of the evasion tactics utilized by current Twitter spammers. To achieve our research goals, we first collect and analyze around 500,000 Twitter accounts and more than 14 million tweets using Twitter API [20]. Then we combine the blacklist, honeypot technique and manual verification strategy to collect one major type of spammers which is relatively more evil than other types of spammers. Then, based on our collected data, we provide a quantitative empirical analysis of the evasion tactics utilized by Twitter spammers and also test the detection rates of three existing state-of-the-art approaches. Through the analysis and examination of these existing works, we show how Twitter spammers can utilize different evasion tactics to evade existing profile-based detection features and content-based detection features. Based on the in-depth analysis of those evasion tactics and Twitter spammers' desire to achieve their malicious goals, we design ten new detection features including graph-based features, neighbor-based features, timing-based features and automation-based features to detect Twitter spammers. Through our experiments on two different datasets, using five different machine learning classifiers, we show that our newly designed features can be used to effectively detect Twitter spammers, especially evasive spammers. In addition, to understand the effectiveness and difficulties of using machine learning features to detect spammers, we also formalize the robustness of 24 detection features that are utilized in the existing work and in our work.

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

- We present the first in-depth empirical analysis of evasion tactics utilized by current Twitter spammers based on a large dataset containing around 500,000 Twitter accounts and more than 14 million tweets.

- We evaluate the detection rates of three state-of-the-art solutions on our collected dataset. Even the best detector still misses detecting around 27% of Twitter spammers and the worst detector misses about half of the spammers.
- Based on our empirical analysis of the evasion tactics and the Twitter spammers' desire to achieve malicious goals, we propose and test our newly designed, more robust detection features. To the best of our knowledge, it is the first work to propose neighbor-based detection features to detect Twitter spammers. According to our evaluation, while keeping an even lower false positive rate, the detection rate by using our new feature set significantly increases to 85%, compared with a detection rate of 51% and 73% for the worst existing detector and the best existing detector, respectively.
- To understand the effectiveness and difficulties of using machine learning features to detect Twitter spammers, we provide a new framework to quantitatively and qualitatively formalize the robustness of 24 detection features that are utilized by the existing work and our work, and categorize them into 14 low-robust features, 6 medium-robust features and 4 high-robust features.

2 Related Work

Due to the rising popularity of Twitter, many studies have been conducted with an aim at studying the topological characteristics of Twitter. Kwa *et al.* [34] has shown a comprehensive, quantitative study of Twitter accounts' behavior, such as the distribution of the number of followers and followings for each account on the entire Twittersphere, and the reciprocity of these relationships. Cha *et al.* [28] have designed diverse metrics to measure Twitter accounts.

In addition, since spam and attacks are so rampant in online social networking sites, Koutrika *et al.* [33] propose techniques to prevent attackers increasing the visibility of an object from fooling the search mechanism to detect tag spam in tagging systems. Benevenuto *et al.* [27, 26] utilized the machine learning technique to identify video spammers in video social networks such as YouTube. Gao *et al.* [30] presented a study on detecting and characterizing social spam campaigns in Facebook.

In terms of Twitter, there has been a few recently published studies on detecting Twitter spammers. Most existing detection work can be classified into two categories. The first category of work, such as [35, 25, 38, 37], mainly utilizes machine learning techniques to classify legitimate accounts and spam accounts according to their collected training data and their selections of classification features. Lee *et al.* [35] deployed a social honeypot for harvesting deceptive spam profiles from Twitter and also utilized the machine learning technique to detect spammers according to their designed features, such as *number of URLs per tweet* and *number of unique @usernames per tweet*. Benevenuto *et al.* [25] also utilizes the machine learning technique using content-based features such as *number of hashtags per word of each tweet* and profile-based features such as *number of followers* and *number of followings*. Wang [38] designs its own features such as *reputation score*, *number of duplicate Tweets* and *number of URLs*. The second category of work, e.g. [31], detects spam accounts by examining whether the URLs or domains of the URLs posted in the tweets are tagged as malicious by the public URL blacklists or domain blacklists. Especially, to collect training data, both [35] and [37] utilize social honey accounts to Twitter Twitter spammers.

Different from existing related work, our work focuses more on analyzing evasion tactics utilized by current Twitter spammers. We design new machine learning features to more effectively detect Twitter spammers. In addition, to better understand the robustness of each detection feature, we quantitatively and qualitatively formalize the robustness of the detection features. Thus, our work is a valuable supplement to existing Twitter spam detection research.

3 Overview

In this section, we state our targeted problems, and introduce our data collection strategies and results.

3.1 Problem Statement

As shown in Section 2, most of the existing studies detect Twitter spam accounts by designing corresponding detection features and utilizing machine learning techniques. However, the arms race nature between the attackers and defenders leads Twitter spammers to evolve or utilize tools to evade existing detection features [12]. Our research goal is to provide the first empirical analysis of the evasion tactics, and then through in-depth analysis of those evasion tactics, we propose new features to detect more Twitter spammers. In addition, to understand the strength of the detection features against evasion, we also analyze the robustness of the detection features.

3.2 Data Collection

In order to achieve our research goal, we need to create a large data set by crawling real Twitter profiles and also identify Twitter spammers in this data set.

To crawl Twitter profiles, we develop a Twitter crawler that taps into Twitter's Streaming API [20]. In order to decrease the effect of the sampling bias [36], we utilize a new crawling strategy rather than simply using the Breath First Search (BFS) sampling technique. Specifically, we first collect 20 seed Twitter accounts from the public timeline [23]. For each of these 20 accounts, we also crawl their followers and followings. We then repeat this process by collecting a new set of 20 seed Twitter accounts from the timeline. For each account that we crawl, we collect the 40 most recent Tweets as well as any other information that Twitter will allow us to collect. Due to the large amount of redirection URLs used in Twitter, we also follow the URL redirection chain to obtain the final destination URL. This resulted in the collection of nearly 500,000 Twitter accounts which posted over 14 million tweets containing almost 6 millon URLs. Details about the crawling information can be seen in Table 1.

Table 1. Twitter accounts crawling information

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

Then, we need to identify Twitter spammers from our crawled dataset. In our work, we focus on those Twitter Spammers that post harmful links to phishing or malware sites, since this type of spammers is more deleterious than other types of spammers. Specifically, we first utilize Google Safe Browsing [10] and Capture-HPC [7] to detect malicious or phishing URLs in the tweets. We define a Tweet that contains at least one malicious or phishing URL as a *Spam Tweet*. For each account, we define its *spam ratio* as the ratio of the number of its *spam tweets* that we detect to the total number of its tweets that we collect. Then, we extract 2,933

Twitter accounts whose spam ratio are higher than 10%. Then, in order to decrease false positives, our group members spend several days on manually verifying all 2,933 accounts and finally obtain 2,060 truly spam accounts.

We acknowledge that our collected Twitter spammers may still contain some bias. However, it is very challenging, even for the Twitter Corporation, to achieve comprehensive ground truth for Twitter spammers. Also, since we focus more on one type of spammers, we realize that the number of spammers in our examination data set is a lower bound on the number of real spammers. We discuss these limitations further in Section 8. However, even for a subset of spammers, we can still use them to analyze the evasion tactics and test the performance of existing work on detecting these spammers.

After collecting our examination data, we are able to show that many of these spam accounts have evolved to be able to evade state-of-the-art detection schemes. We will show and perform an in-depth analysis of the evasion tactics utilized by spammers in Section 4. Then, according to the analysis, we will design new detection features to counter these tactics in Section 5 and formalize the robustness of the detection features in Section 6. Finally, in Section 7, we show that our newly designed features can be an effective supplement to the existing detection approaches.

4 Analyzing Evasion Tactics

This section will describe the evasive tactics that spammers are using to evade existing machine learning detection schemes. Then, we validate these tactics by both showing some case studies and examining the existing work on our collected data set.

4.1 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.

Profile-based Feature Evasion Tactics: 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*[1] [35, 37] and *reputation score* [38] are built based on this number. To evade these features or break Twitter's 2,000 Following Limit Policy[2] [15], 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

[1] It is the ratio of the number of an account's following to its followers.

[2] According to this policy, if the number of following of an account is exceeding 2,000, this number is limited by the number of the account's followers.

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 contributed 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. [37]). 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 [3].

Content-based Feature Evasion Tactics: Another common indicator of disclosing spam accounts is the content of a suspect account's Tweets. As discussed in Section 1, a majority of spam accounts make profits by alluring legitimate users to click the malicious URLs posted in the spam tweets. Those malicious URLs can direct users to websites that may 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 work such as [35, 37, 38]. In addition, since many spammers repeat posting the same or similar malicious tweets in order to increase the probability of successfully alluring legitimate users' visits, especially with the help of some automation tweeting tools, their published tweets shows strong homogeneous characteristics. In this way, many existing approaches design content-based features such as *tweet similarity* [35, 37] and *duplicate tweet count* [38] to detect spam accounts. To evade such content-based detection features, spammers mainly utilize the tactics including: mixing normal tweets and posting heterogenous tweets.

Mixing Normal Tweets: Spammers can utilize this tactic to evade content-based features such as *URL ratio*, *unique URL ratio*, *hashtag ratio* [35, 38]. These normal tweets without malicious URLs may be hand-crafted or obtained from arbitrary users' tweets or consisted of totally 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.

4.2 Validation of Evasion Tactics

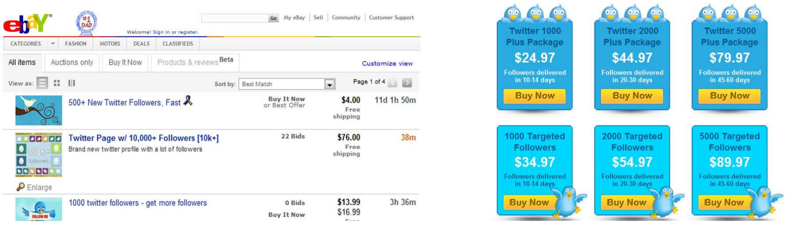
In this section, we aim to validate the four evasion tactics described in the previous section by showing real case studies and public services/tools that can be utilized by the spammers. We also implement existing detection schemes [35, 37, 38] and evaluate them on our collected examination data set. 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 Section 4.1, 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 2 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 2. 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.



500+ New Twitter Followers, Fast — Buy It Now or Best Offer — $4.00 Free shipping — 11d 1h 50m

Twitter Page w/ 10,000+ Followers [10k+] — Brand new twitter profile with a lot of followers — 22 Bids — $76.00 Free shipping — 38m

1000 twitter followers - get more followers — 0 Bids Buy It Now — $13.99 $16.99 — 3h 36m

Twitter 1000 Plus Package $24.97 Followers delivered in 10-14 days

Twitter 2000 Plus Package $44.97 Followers delivered in 20-30 days

Twitter 5000 Plus Package $79.97 Followers delivered in 45-60 days

1000 Targeted Followers $34.97 Followers delivered in 10-14 days

2000 Targeted Followers $54.97 Followers delivered in 20-30 days

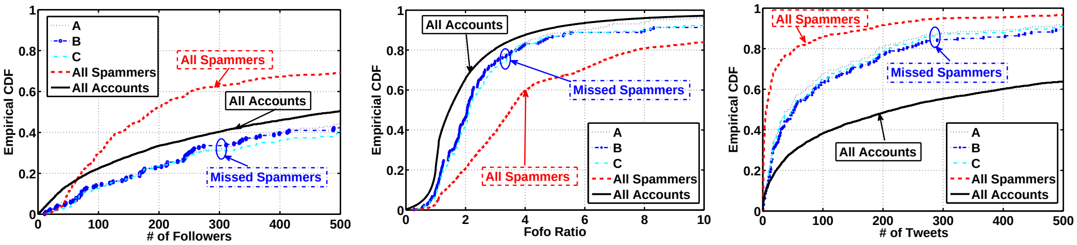
5000 Targeted Followers $89.97 Followers delivered in 45-60 days

(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 [35, 37, 38], *all accounts* (around 500,000 collected accounts), and *all spammers* (2,060 identified spammers). To better show the results, we label the results from [38] as A, [35] as B and [37] 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 [3] and Twitter API [20] 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.

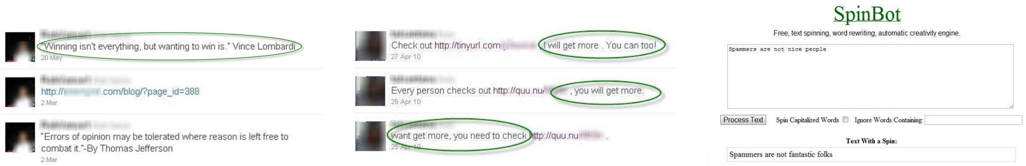


(a) Number of followers (b) Fofo Ratio (b) Number of tweets

Fig. 2. Profile-based feature examination on three existing detection work

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 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.


[redacted]
"Winning isn't everything, but wanting to win is." Vince Lombardi
20 May

[redacted]
http://[redacted].com/blog/?page_id=388
2 Mar

[redacted]
"Errors of opinion may be tolerated where reason is left free to combat it."-By Thomas Jefferson
2 Mar

[redacted]
Check out http://tinyurl.com[redacted] I will get more . You can too!
27 Apr 10

[redacted]
Every person checks out http://quu.nu/[redacted] , you will get more.
26 Apr 10

[redacted]
want get more, you need to check http://quu.nu[redacted] .
25 Apr 10

SpinBot

Free, text spinning, word rewriting, automatic creativity engine.

```
Spammers are not nice people
```

Process Text   Spin Capitalized Words □   Ignore Words Containing:

Text With a Spin:

Spammers are not fantastic folks

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

Fig. 3. Case studies for content-based feature evasion tactics

From the above analysis and validation, we can find that Twitter spam accounts are indeed evolving to evade existing detection methods to increase their lifespan.

5 Designing New Features

In this section, to counter spammers' evasion tactics, we propose several new and more robust detection features. A robust feature should either be difficult or expensive for the malicious entity to evade: a feature is difficult to evade if it requires a fundamental change in the way a malicious entity performs its malicious deeds; a feature is expensive to evade if the evasion requires much money, time or resources. On the basis of spammers' special characteristics, we design 10 new detection features including 3 Graph-based features, 3 Neighbor-based features, 3 Automation-based features and 1 Timing-based feature, which will be described in detail in the following sections.

5.1 Graph-based Features

If we view each Twitter account i as a node and each follow relationship as a directed edge e, then we can view the whole Twittersphere as a directed graph $G = (V, E)$. Even though the spammers can change their tweeting or following behavior, it will be difficult for them to change their positions in this graph. According to this intuition, we design three graph-based features: local clustering coefficient, betweenness centrality, and bi-directional links ratio.

Local Clustering Coefficient: The local clustering coefficient [11] for a vertex is the proportion of links between the vertices within its neighborhood divided by the number of links that could possibly exist between them. This metric can be utilized to quantify how close a vertex's neighbors are to being a clique. For each vertex v in the Twitter graph, its local clustering score can be computed by Eq. (1), where K_v is the sum of the indegree and outdegree of the vertex v, and $|e^v|$ is the total number of edges built by all v's neighbors.

$$LC(v) = \frac{2|e^v|}{K_v \cdot (K_v - 1)} \tag{1}$$

Since legitimate users usually follow accounts whose owners are their friends, colleagues or family members, these accounts are likely to have a relationship with each other. However, since spammers usually blindly follow other accounts, these accounts usually do not know each other and have a looser relationship among them. Thus, compared with the legitimate accounts, Twitter spammers will have smaller local clustering coefficient.

Betweenness Centrality: Betweenness centrality [4] is a centrality measure of a vertex within a graph. Vertices that occur on many shortest paths between other vertices have a higher betweenness than those that do not. In a directed graph, betweeness centrality of each vertex v can be computed by Eq. (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.

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

This metric reflects the position of the vertex in the graph. Nodes that occur in many shortest paths have higher values of betweenness centrality. A Twitter spammer will typically use a shotgun approach to finding victims, which means it will follow many 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 between any victim following of the spam account and any other victim following, through the spam account. Thus, the betweenness centrality of the spammer will be high.

An informed spammer would be able to carefully choose his victims and force their betweenness centrality and clustering coefficient values to be similar to those of a normal Twitter user. However, this not only requires more time and skill to implement, but also limits the potential victims of spammers. A spammer that is forced to pick and choose specific accounts will be much less effective than the one that can randomly choose his victims.

Bi-directional Links Ratio: If two accounts follow with each other, we consider them to have a bidirectional link between each other. The number of bi-directional links of an account reflects the reciprocity between an account and its followings. Since Twitter spammers usually follow a large number of legitimate accounts and cannot force those legitimate accounts to follow back, the number of bi-directional links that a spammer has is low. On the other hand, a legitimate user is likely to follow his friends, family members, or co-workers who will follow this user back. Thus, this indication can be used to distinguish spammers. However, Twitter spammers could evade this by following back their followers. Thus, we create another feature named *bi-directional links ratio* (R_{bilink}), which can be computed in Eq. (3).

$$R_{bilink} = \frac{N_{bilink}}{N_{fing}} \tag{3}$$

where N_{bilink} and N_fing denote the number of bi-directional links and the number of followings. The intuition behind this feature is that even though the spammers can increase the value of N_{bilink} through following back their followers or obtaining "following-backs" from other accounts, compared with their high values of N_{fing}, their values of R_{bilink} will be relatively difficult to increase to be comparable with that of legitimate accounts. Although this feature still can be evaded, the spammers need to pay more to evade this feature.

5.2 Neighbor-based Features

Due to the fact that spammers can control their own behavior but can not control their following accounts' behavior, in this section, we design three neighbor-based features to distinguish Twitter spammers and legitimate accounts: average neighbors' followers, average neighbors' tweets, and followings to median neighbors' followers.

Average Neighbors' Followers: Average neighbors' followers, denoted as A_{nfer}, of an account v represents the average number of followers of this account's followings, which can be computed with Eq.(4).

$$A_{nfer}(v) = \frac{1}{|N_{fing}(v)|} \cdot \sum_{u \in N_{fing}(v)} N_{fer}(u) \tag{4}$$

where N_{fer} and N_{fing} denote the number of followers and followings, respectively. Since an accounts' follower number usually reflects this account's popularity or reputation, this feature reflects the quality of the choice of friends of an account. It is obvious that legitimate accounts intend to follow the accounts who have higher quality unlike the spammers. Thus, the average neighbors' followers of legitimate accounts are commonly higher than that of spammers.

Average Neighbors' Tweets: Similar to the average neighbors' followers, since an account's tweet number could also reflect this account's quality, we design another feature, named *average neighbors' tweets*, which is the average number of tweets of this account's following accounts. Note that these two features can be evaded by following popular Twitter accounts (seen in Section 6). We also design another relatively robust neighbor-based detection feature, named followings to median neighbors' followers.

Followings to Median Neighbors' Followers: To extract this feature, we first calculate the median number of an account's all following accounts' follower numbers, denoted as M_{nfer}. Then, the value of the followings to median neighbors' followers of an account, denoted as R_{fing_mnfer}, can be computed by the ratio of this account's following number to that median number, as shown in Eq.(5).

$$R_{fing\_mnfer} = \frac{N_{fing}}{M_{nfer}} \tag{5}$$

Since the spammers can not guarantee the quality of the accounts they follow, their values of M_{nfer} are typically small. Thus, due to spammers' large numbers of followings, spammers' values of R_{fing_mnfer} will be also high. For the legitimate accounts, to show the analysis of this feature, we divide them into two different types: common accounts (legitimate accounts without large numbers of followers) and popular accounts (legitimate accounts with large numbers of followers). For the first type of accounts, they may also just follow their friends which leads to a small value of M_{nfer}. However, since their following numbers are also not high, common accounts' values of R_{fing_mnfer} are not high. For the popular accounts who are are usually celebrities, famous politicians, or professional institutions, they will usually choose accounts who are also popular to follow. In this way, these accounts' values of M_{nfer} will be high, leading to low values of R_{fing_mnfer}. From the above analysis, we can find that spammers will have higher values of this feature than that of legitimate accounts. In addition, since we use the median value rather than the mean, it will be very difficult for spammers to increase their values of M_{nfer} by following a few very popular accounts. Thus, this feature is difficult to be evaded.

5.3 Automation-based Features

Due to the large cost of manually managing a large number of spam accounts, many spammers choose to create a custom program using Twitter API to post spam tweets. Thus, we also design three automation-based features to detect spammers: API[3] ratio, API URL ratio and API Tweet Similarity.

API Ratio: API ratio is the ratio of the number of tweets with the tweet source of "API" to the total number of tweet count. As existing work [29] shows, many bots choose to use API to post tweets, so a high API ratio implies this account is more suspicious.

API URL Ratio: API URL ratio is the ratio of the number of tweets containing a URL posted by API to the total number of tweets posted by API. Since it is more convenient for spammers to post spam tweets using API, especially when spammers need to manage a large amount of accounts, as discussed in Section 4. Thus, a higher API URL ratio of an account implies that this account's tweets sent from API are more likely to contain URLs, making this account more suspicious.

[3] The tweet source of the tweets sent by unregistered third-party applications in Twitter will be labeled as "API" rather than the application names like "TweetDeck" [18]. Thus, in this paper, we use "API" to refer those unregistered third-party applications.

API Tweet Similarity: Spammers can use tricks to evade the detection feature of *tweet similarity* as described in Section 4 and still choose to use API to automatically post malicious tweets. Thus, we also design *API tweet similarity*, which only compute the similarity of those tweets posted by API. Thus, a higher API tweet similarity of an account implies that this account is more suspicious.

5.4 Timing-based Features

Similar to other timing-based features such as *tweeting rate* presented in [25], we also design another feature named *following rate*.

Following Rate: Following rate reflects the speed at which an account follows other accounts. Since spammers will usually follow many users in a short period of time, a high following rate of an account indicates that the account is likely a spam account. Since it is difficult to collect the time when an account follows another account, we use an approximation to calculate this feature. Specifically, we use the ratio of an account's following number to the age of the account at the time the following number was recorded. Note that this feature can still be evaded with cost and we use the approximate method to calculate this feature due to practical reasons. We analyze the robustness of this feature in Section 6, and discuss this limitation in Section 8.

After designing these new features, we first formalize the robustness of most of the existing detection features and our designed features in Section 6. Then, we combine some existing effective features and our features to build a new machine learning detection scheme and evaluate it based on our dataset in Section 7.

6 Formalizing Feature Robustness

In this section, to deeply analyze how to design effective features to detect Twitter spammers along with their evolution, we formalize the robustness of the detection features.

6.1 Formalizing the Robustness

Before analyzing the robustness of each feature in detail, we first build a quantitative model to define the robustness of the detection features. Due to the dual-puropse nature of the spammers – avoiding detection and achieving malicious goals, the robustness of each feature F, denoted as $R(F)$, can be viewed as the tradeoff between the spammers' cost $C(F)$ to avoid the detection and the profits $P(F)$ by achieving malicious goals. Thus, the robustness of each feature can be computed by Eq. (6).

$$R(F) = C(F) - P(F) \tag{6}$$

Then, if the cost of evading the detection feature is much higher than the profits, this feature is relatively robust. To quantify the evasion, we use T_F to denote the threshold for spammers to meet to evade each detection feature F.

From the viewpoints of Twitter spammers, the cost to evade the detection mainly includes: money cost, operation cost and time cost. The money cost is mainly related to obtaining followers. We use C_{fer} to denote the cost for the spammer to obtain one follower. The operation cost is mainly related to posting tweets or following specific accounts. We use C_{twt} and C_{follow} to denote the cost for a spammer to post one tweet or follow one specific account. Spammers' profits are achieved by attracting legitimate accounts' attention. Thus, Twitter spammers' profits can be measured by the number of followings that they can support and the number of spam tweets that they can post. And we use P_{fing} and P_{mt} to denote the profit of supporting one following account, obtaining one following back and posting one spam tweet, respectively. Let N_{fing} and N_{mt} denote the number of accounts that a spammer desires to follow and the number of malicious tweets that the spammer desires to post.

In this section, we show our analysis of the robustness for the following 6 categories of features: profile-based features, content-based features, graph-based features, neighbor-based features, timing-based features and automation-based features. The summary of these features can be seen in Table 3.

Table 3. Detection Feature Robustness

Index	Category	Feature	Used in Work	Robustness
F_1	Profile	the number of followers (N_{fer},)	[38]	Low
F_2 (+)	Profile	the number of followings (N_{fing})	[37], [38], ours	Low
F_3 (+)	Profile	fofo ratio (R_{fofo})	[35], [37], ours	medium
F_4	Profile	reputation (Rep)	[38]	medium
F_5 (+)	Profile	the number of tweets (N_{twt})	[37], ours	Low
F_6 (+)	Profile	age	[35], ours	High
F_7 (+)	Content	URL ratio (R_{URL})	[35], [37], [38], ours	Low
F_8 (+)	Content	unique URL ratio	[35], ours	Low
F_9	Content	hashtag(#) ratio	[38]	Low
F_{10}	Content	reply(@) ratio	[35], [38]	Low
F_{11} (+)	Content	tweet similarity (T_{sim})	[35], [37], ours	Low
F_{12}	Content	duplicate tweet count	[38]	Low
F_{13}	Graph	number of bi-directional links (N_{bilink})	[35]	Low
F_{14} (*)	Graph	bi-directional links ratio (R_{bilink})	ours	medium
F_{15} (*)	Graph	betweenness centrality (BC)	ours	High
F_{16} (*)	Graph	clustering coefficient (CC)	ours	High
F_{17} (*)	Neighbor	average neighbors' followers (A_{nfer})	ours	Low
F_{18} (*)	Neighbor	average neighbors' tweets (A_{ntwt})	ours	Low
F_{19} (*)	Neighbor	followings to median neighbors' followers (R_{fing_mnfer})	ours	High
F_{20} (*)	Timing	following rate (FR)	ours	Low
F_{21} (+)	Timing	tweet rate (TR)	[35], ours	Low
F_{22} (*)	Automation	API ratio (R_{API})	ours	medium
F_{23} (*)	Automation	API URL ratio (R_{API_URL})	ours	medium
F_{24} (*)	Automation	API Tweet Similarity (T_{api_sim})	ours	medium

Robustness of Profile-based Features: As described in Section 4, spammers usually evade this type of detection features by obtaining more followers. According to Eq.(6), the robustness of the detection feature *fofo ratio*(F_3), which is a representative feature of this type, can be computed by Eq.(7).

$$R(F_3) = \frac{N_{fing}}{T_{F_3}} \cdot C_{fer} - N_{foing} \cdot P_{fing} \quad (T_{F_3} \geq 1) \tag{7}$$

From Table 2, we can find that C_{fer} can be very cheap, which could be much smaller compared with the P_{foing}. Even when the spammers who desire to follow 2,000 accounts, which is restricted by Twitter's 2,000 Following Limit Policy, they just need to spend $50 to breakthrough that policy. Thus, this feature can be evaded by spending little money. Similar conclusions can be drawn for the features F_1, F_2 and F_4.

For feature F_6, since the age of an account is determined by the time when the account is created, which can not be changed or modified by the spammers, this feature is relatively hard to evade. It could also be evaded if the spammers can use some tricks to obtain Twitter accounts that were registered in Twitter a long time ago. But unlike obtaining followers, obtaining a specific Twitter account will be very expensive. For example, the bid value of purchasing a Twitter account that steadily has over 1,390 followers is $1,550 [19].

Since *number of tweets*(F_5) is related to several content-based features, we show the analysis of this feature in the next section.

Robustness of Content-based Features: As shown in Table 3, content-based features can be divided into two types: signature-based features (F_7, F_8, F_9, and F_{10}) based on special terms or tags in the tweets and similarity-based features (F_{11}, and F_{12}) based on the similarity among the tweets. As discussed in Section 4, both types of features can be evaded by automatically posting non-signature tweets or diverse tweets. Also, by using these tactics, the spammers can evade the feature of the number of tweets (F_5).

Without the loss of the generality, we use the analysis of the robustness of the *URL_ratio* (F_7) to represent the analysis of this type of features. Similar as Eq.(7), if a spammer needs to post N_{mt} tweets with the malicious URLs, the robustness for F_7 can be computed by Eq.(8).

$$R(F_7) = \frac{N_{mt}}{T_{F_7}} \cdot C_{twt} - N_{mt} \cdot P_{mt} \quad (T_{F_7} \leq 1) \tag{8}$$

From Eq.(8), we can find that if spammers utilize software such as AutoTwitter [3] and Twitter API [20] to automatically post tweets, C_{twt} will be small. So even when we set a small value of T_{F_7} (e.g., 0.1)d, compared with the big profits of successfully alluring the victims to click the malicious URLs, the cost is still small.

Robustness of Graph-based Features: For the graph-based features, we can divide them into two types: reciprocity-based features (F_{13} and F_{14}) based on the number of the bi-directional links and position-based features (F_{15} and F_{16}) based on the position in the graph. If we denote C_{BiLink} as the cost to obtain one bi-directional link, then the robustness of F_{13} and F_{14} can be computed in Eq. (9) and (10).

$$R(F_{13}) = T_{F_{13}} \cdot C_{BiLink} \tag{9}$$

$$R(F_{14}) = T_{F_{14}} \cdot N_{fing} \cdot C_{BiLink} \tag{10}$$

Since it is impractical to set a high bi-directional link threshold to distinguish legitimate accounts and spammers, the value of $T_{F_{13}}$ could not be high. Meanwhile, when $T_{F_{13}}$ is small, spammers can obtain bi-directional links by following their followers. Thus, the C_{BiLink} is also not high. Thus, from Eq. 9, we can find that $R(F_{13})$ is not big. For feature F_{14}, since the average of the bi-directional links ratio is 22.1% [34] and the spammers usually have a large value of N_{fing}, the spammers need to obtain much more bidirectional links to show a normal bi-directional links ratio. Even though this feature could be evaded by following spammers' followers, due to the difficulties of forcing those accounts to follow spammers back, it will cost much to evade this feature.

For the position-based features, since spammers usually blindly follow legitimate accounts, which may not follow those spammers back, it will be difficult for spammers to change their positions in the whole social network graph. Similarly, spammers can neither control the accounts they followed to build social links with each other. In this way, it is difficult for spammers to change their values of the graph metrics, thus to evade graph-based features.

Robustness of Neighbor-based Features: The first two neighbor-based features reflect the quality of an account's friend choice, which has been discussed in Section 5. If we use N_{follow} to denote the number of popular accounts (the accounts who have very big follower numbers) 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.(11).

$$R(F_{17}) = N_{follow} \cdot C_{follow} \tag{11}$$

Since there are many popular accounts with very big 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 we use the median not the mean of the neighbors' followers, they need to follow around half of N_{fing} popular accounts to evade this feature. With a consideration of spammers' big values of N_{fing}, the cost will be very high and the profit will be decreased dramatically for the spammers to evade this feature. So, feature F_{19} is relatively difficult to evade.

Robustness of Timing-based Features: The timing-based features are related to spammers' update behavior. Although the profits may drop, when spammers decrease their following or tweeting rate, since these two features can be totally controlled by the spammers, the cost will be low. Thus, feature F_{20} and F_{21} are relatively easy to evade.

Robustness of Automation-based Features: As discussed in Section 5, in order to more efficiently achieve the malicious goals, many Twitter spammers will control multiple spam accounts to spread the spam tweets. Similar as the discussion of the content-based features, if the spammers want to evade automation-based features, they may also need to use software to manage those spam accounts to automatically post tweets. Due to the simplicity and convenience of using API, many spammers use API to post tweets.

If we use C_{twt_web} and C_{twt_api} to denote the cost of using web and API to post one tweet, it is obvious that if the spammers desire to post a large amount of tweets, the cost of posting by web will be much higher than that of using API, (i.e., $C_{twt\_web} \gg C_{twt\_api}$). If a spammer desires to use API to post spam tweets on N_{spam} spam accounts, then the robustness of feature F_{22} can be computed as Eq. (12).

$$R(F_{22}) = N_{spam} \cdot [\frac{N_{mt}}{T_{F_{22}}} \cdot (1 - T_{F_{22}}) \cdot C_{twt\_web} + N_{mt} \cdot C_{twt\_api}] - N_{spam} \cdot N_{mt} \cdot P_{mt} \tag{12}$$

Since few legitimate accounts would use API to post tweets, $T_{F_{22}}$ can be set as very small. In this way, since $C_{twt\_web} \gg C_{twt\_api}$, if a spammer wants to control many spammer accounts and post a large amount of tweets, the cost will be relatively high. The conclusions for the rest of this type of features are similar.

It is noticeable that only using feature F_{22} will bring some false positives since legitimate accounts may also use API to post tweets. However, if we combine to use the features of F_{22}, F_{23}, and F_{24} together, it will decrease those false positives, since few legitimate accounts would use API to post very similar tweets with URLs as spammers do.

In summary, through the above analysis, we can categorize the robustness of these detection features into the following three scales: low, medium, and high. The summary of this information can be seen in Table 3.

7 Evaluation

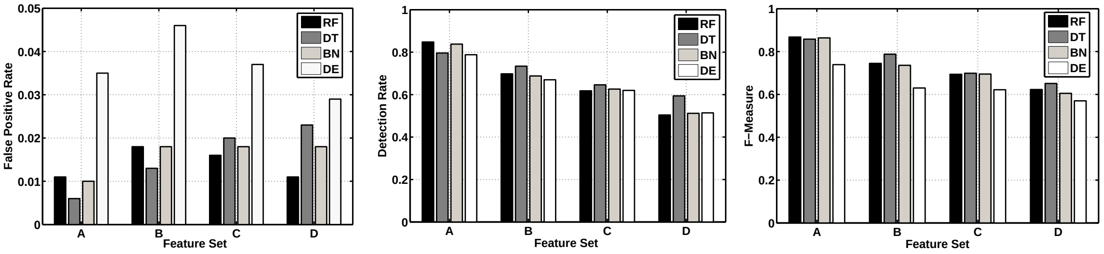
In this section, we will evaluate the performance of our designed machine learning feature set including: 8 existing effective features marked with (+) and 10 newly designed features marked with (*) in Table 3.

We evaluate the feature set by implementing machine learning techniques on two different data sets: Data set I and Data set II. Data set I consists of 5,000 accounts without any spam tweets and 500 identified spammers, which are randomly selected from our crawled dataset described in Section 3. To decrease the effects of sampling bias and show the quality of our detection feature schema without using URL analysis as ground truth, we also crawled another 35,000 Twitter accounts and randomly selected 3,500 accounts to build another data set, denoted as Data set II.

7.1 Evaluation on Data set I

In this section, based on Data set I, we make three kinds of experiments to evaluate our machine learning feature set: performance comparison, feature validation and learning curve.

Performance Comparison: In this experiment, we compare the performance of our work with three existing approaches[4]: [35], [37] and [38]. We conduct our evaluation by using four different machine learning classifiers: *Random Forest (RF)*, *Decision Tree (DT)* , *Bayes Net (BN)* and *Decorate (DE)*. To better show the results, we label our method as *A*, [35] as *B*, [37] as *C*, and [38] as *D*. For each machine learning classifier, we use *ten-fold cross validation* to compute three metrics: *false positive rate*, *detection rate*, and *F-measure*[5].



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

Fig. 4. Performance comparison with the existing approaches

As seen in Fig. 4, our approach outperforms existing work. Specifically, from Fig. 4(a), we can find that the false positive rates of our work under three machine learning classifiers (RF, DT and BN), are the lowest and the false positive rate of our work under the other classifier (DE) is the second lowest. Especially, under the decision tree classifier (DT), which is a standard and prevalent machine learning classifier, the false positive rate of our work (0.5%) is less than half of the best other existing approach (B) and a quarter of the worst one (D). From Fig. 4(b), we can find that the detection rates of our work under all four machine learning classifiers are the highest. In particular, the detection rate of our work (85%) is significantly higher than the detection rate of 51% for the worst detector (D) and

[4] The features used in these three approaches can be seen in Table 3.

[5] F-measure [9] is a measure with the consideration of both precision and recall.

the detection rate of 73% for the best other existing detector (B). We also evaluate our feature set based on the metric of F-measure [9]. Fig. 4(c) shows that under all four classifiers, F-measure scores of our approach are the highest. The above results validate that our new feature set is more effective to detect Twitter spammers.

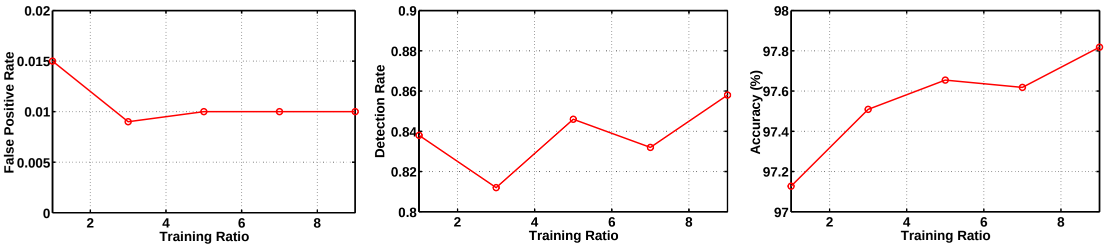
Through these three figures, we can also observe that the performance of [35] and [37] is better than that of [38]. That is mainly because both [35] and [37] utilize the feature of *tweet similarity*, and [38] only uses the feature of *duplicate tweet count*. Since many spammers post tweets with similar terms but different combinations rather than simply repeatly posting the same tweet, the feature of *tweet similarity* is much more effective than *duplicate count*. Also, [35] utilizes a graph-based feature (*number of bi-directional links*) and a timing-based feature *tweet rate*, leading its performance to be better than that of [37].

Feature Validation: To further validate the effectiveness of our newly designed features, we make the comparison of the performance of two feature sets. The first one consists of the features used in the previous experiment without our newly designed 10 features. The second one consists of all features used in the previous experiment. From Table 4, we can see that for each classifier, with the addition of our newly designed features, the detection rate increases over 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 4. Comparison Without and With New Features

	Without Our Features			With Our Features		
Algorithm	False Positive Rate	Detection Rate	F-Measure	False Positive Rate	Detection Rate	F-Measure
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	0.644	0.730	0.023	0.784	0.777

Learning Curve: In this experiment, we show the steadiness of our feature set by varying the training ratio, which is the ratio of the number of training data to the number of testing data. Specifically, we use Decorate to evaluate our feature set based on the metrics of *false positive rate*, *detection rate*, and *accuracy*. The results can be seen in Fig. 5.



(a) False Positive

(b) Detection Rate

(c) Accuracy

Fig. 5. Performance comparison with the existing approaches

As Fig. 5(a) shows, even when the training ratio varies, our approach can obtain a low false positive rate, never rising above 1.5%. Fig. 5(b) shows that our approach can steadily detect over 80% spammers under different training ratios. From Fig. 5(c), we can see that the total accuracy can remain higher than 97%, even the training ratio is small (e.g. 1:1).

7.2 Evaluation on Dataset II

In this section, to decrease possible effect of sampling bias, we evaluate the effectiveness of our detection feature set by testing it on another data set containing 3,500 unclassified Twitter accounts. Our goal of the evaluation on another crawled dataset is to test the actual operation and user experience without the ground truth from URL analysis by computing the Bayesian detection rate [24] – the probability of actually being at least a suspicious spammer, whenever an account is reported by the detection system.

Specifically, we use Data set I, which has been labeled, as the training data set, and Data set II as the testing data. Then, based on our detection feature set, we use BayesNet classifier to predict spammers on Data set II. This result can be seen in Table 5.

Table 5. Classifier Effectiveness

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

When we manually investigated those 70 accounts that were predicted as spammers, we found 37 real spammers, 25 promotional advertisers[6] and only 8 real false positives. In this case, we have a high Bayesian detection rate of 88.6% (62/70). Then, we further investigate these 8 false positive Twitter accounts. We find that all of them have odd behavior, but do not appear to have clear malicious intentions. Specifically, 6 of them are actively tweeting about only one topic. The other 2 have posted very few tweets, yet have a large number followings with a high ratio of followings to followers. Also, we examined the URLs that these 37 verified spammers posted to Twitter, and we found 17 of them posted malicious URLs according to the Google Safe Browsing blacklist.

8 Limitation and Future Work

Due to practical limitations, we can only crawl a portion of the whole Twittersphere and our crawled data set may still have sampling bias. However, collecting an ideal large data set from Twitter, a real and dynamic OSN, without any bias is almost an impossible mission.

In addition, it is challenging to achieve comprehensive ground truth for Twitter spammers. Also, since we collect one major type of spammers, the number of our identified spammers is a lower bound of them in our dataset. However, even for a subset of spammers, we can find that they are evolving to evade detection. And our evaluation validates the effectiveness

[6] Since some consider Promotional Advertisements to be spam and others do not, we label these accounts as another category. At least, These accounts are very suspicious.

of our newly designed features to detect these spammers. We also acknowledge that some identified spam accounts may be compromised accounts. However, since these accounts still behave fairly maliciously in their recent histories and are dangerous to the Twittersphere, it is also meaningful to detect them.

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, we extract the approximate values of these two features by using a sampling technique that allowed us to compute these metrics piece-by-piece. However, precisely estimating the values of such graph metrics on large graphs such as the one we have crawled is very challenging and a hot research issue, which is out of scope of this work. Also, since we can not extract the exact time when an account follows another, we use an approximation to calculate the feature of *following rate*. Even though this feature may be not perfectly accurate, an approximate value of this feature can still reflect how radically an account increases its following number.

For future work, to overcome those limitations, we will design better crawling strategies and crawl more data. We plan to design more robust features, evaluate our machine learning detection scheme on larger data sets, and work directly with Twitter. We also plan to broaden our targeted type of spammers, so that we can perform a deeper analysis on the evasion tactics by different types of spammers. We also plan to make more quantitative models for the analysis of the robustness of the detection features by deeper analyzing the envision tactics.

9 Conclusion

In this paper, we design new and more robust features to detect Twitter spammers based on an in-depth analysis of the evasion tactics utilized by Twitter spammers. Through the analysis of those evasion tactics and the examination of three state-of-the-art solutions, we design several new features. In addition, in terms of spammers dual objectives – staying alive and achieving malicious goals, we also formalize the robustness of detection features for the first time in the literature. Finally, according to our evaluation, while keeping an even lower false positive rate, the detection rate by using our new feature set increases over 10% than all existing detectors under four different prevalent machine learning classifiers.

References

1. A new look at spam by the numbers. `http://scitech.blogs.cnn.com/`.
2. Acai Berry spammers hack Twitter accounts to spread adverts. `http://www.sophos.com/blogs/gc/g/2009/05/24/acai-berry-spammers-\\hack-twitter-accounts-spread-adverts/`.
3. Auto Twitter. `http://www.autotweeter.in/`.
4. Betweenness Centrality. `http://en.wikipedia.org/wiki/Centrality`.
5. Botnet over Twitter. `http://compsci.ca/blog/`.
6. Buy a follower. `http://http://buyafollower.com/`.
7. Capture HPC. `https://projects.honeynet.org/capture-hpc`.
8. Costolo: Twitter Now Has 190 Million Users Tweeting 65 Million Times A Day. `http://techcrunch.com/2010/06/08/twitter-190-million-users/`.
9. F-measure. `http://en.wikipedia.org/wiki/F1_score`.
10. Google Safe Browsing API. `http://code.google.com/apis/safebrowsing/`.
11. Local Clustering Coefficient. `http://ikipedia.org/wiki/Clustering_coefficient#Local_clustering_coefficienty`.
12. Low-Priced Twitter Spam Kit Sold on Underground Forums. `http://news.softpedia.com/news/Low-Priced-Twitter-Spam-Kit-Sold-on-Underground-Forums-146160.shtml`.

13. New Koobface campaign spreading on Facebook. http://community.websense.com/blogs/securitylabs/archive/2011/01/14/new-koobface-campaign-spreading-on-facebook.aspx.
14. Report Twitter Spam. http://twitter.com/spam/status/914219940.
15. The 2000 Following Limit Policy On Twitter. http://twittnotes.com/2009/03/2000-following-limit-on-twitter.html.
16. The Twitter Rules. http://help.twitter.com/entries/18311-the-twitter-rules.
17. Tweet spinning your way to the top. http://blog.spinbot.com/2011/03/tweet-spinning-your-way-to-the-top/.
18. TweetDeck. http://www.tweetdeck.com/.
19. Twitter account for sale. http://www.potpiegirl.com/2008/04/buy-sell-twitter-account/.
20. Twitter API in Wikipedia. http://apiwiki.twitter.com/.
21. Twitter follower exchange. http://www.tfollower.com/.
22. Twitter phishing hack hits BBC, Guardian and cabinet minister. . http://www.guardian.co.uk/technology/2010/feb/26/twitter-hack-spread-phishing.
23. Twitter Public Timeline. http://twitter.com/public_timeline.
24. Stefan Axelsson. The base-rate fallacy and its implications for the difficulty of intrusion detection. In *In Proceedings of the 6th ACM Conference on Computer and Communications Security*, pages 1–7, 1999.
25. F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. Detecting Spammers on Twitter. In *Collaboration, Electronic messaging, Anti-Abuse and Spam Confference (CEAS)*, 2010.
26. Fabrcio Benevenuto, Tiago Rodrigues, Virgflio Almeida, Jussara Almeida, and Marcos Gonalves. Detecting Spammers and Content Promoters in Online Video Social Networks. In *ACM SIGIR Conference (SIGIR)*, 2009.
27. Fabricio Benevenuto, Tiago Rodrigues, Virgflio Almeida, Jussara Almeida, Chao Zhang, and Keith Ross. Identifying Video Spammers in Online Social Networks. In *Int'l Workshop on Adversarial Information Retrieval on the Web (AirWeb'08)*, 2008.
28. Meeyoung Cha, Hamed Haddadi, Fabricio Benevenuto, and Krishna P. Gummadi. Measuring User Influence in Twitter: The Million Follower Fallacy. In *Int'l AAAI Conference on Weblogs and Social Media (ICWSM)*, 2010.
29. Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia. Who is Tweeting on Twitter: Human, Bot, or Cyborg? In *Annual Computer Security Applications Conference (ACSAC'10)*, 2010.
30. H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Zhao. Detecting and Characterizing Social Spam Campaigns. In *Proceedings of ACM SIGCOMM IMC (IMC'10)*, 2010.
31. C. Griery, K. Thomas, V. Paxsony, and M. Zhangy. @spam: The Underground on 140 Characters or Less. In *ACM Conference on Computer and Communications Security (CCS)*, 2010.
32. Daniel Ionescu. Twitter Warns of New Phishing Scam. http://www.pcworld.com/article/174660/twitter_warns_of_new_phishing_scam.html.
33. G. Koutrika, F. Effendi, Z. Gyongyi, P. Heymann, and H. Garcia-Molina. Combating spam in tagging systems. In *Int'l Workshop on Adversarial Information Retrieval on the Web (AIRWeb'07)*, 2007.
34. H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a Social Network or a News Media? In *Int'l World Wide Web (WWW '10)*, 2010.
35. K. Lee, J. Caverlee, and S. Webb. Uncovering Social Spammers: Social Honeypots + Machine Learning. In *ACM SIGIR Conference (SIGIR)*, 2010.
36. J. Leskovec and C. Faloutsos. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2006.
37. G. Stringhini, S. Barbara, C. Kruegel, and G. Vigna. Detecting Spammers On Social Networks. In *Annual Computer Security Applications Conference (ACSAC'10)*, 2010.
38. A. Wang. Don't follow me: spam detecting in Twitter. In *Int'l Conferene on Security and Cryptography (SECRYPT)*, 2010.