Empirical Evaluation and New Design for Fighting Evolving Twitter Spammers

Chao Yang, Robert Harkreader, and Guofei Gu, *Member, IEEE*

***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. 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 Inc. 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 analyze 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 is also significantly higher than that of existing work. 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.**

***Index Terms*—Online Social Network Websites, Spam, Twitter, Machine Learning**

I. 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 [2].

Spammers have utilized Twitter as a new platform to achieve their malicious goals such as sending spam [3], spreading malware [4], hosting botnet command and control (C&C) channels [5], and performing other illicit activities [6]. 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 [7]. In May of 2009, many innocent users' accounts on Twitter were hacked to spread advertisements [3]. In February of 2010, thousands of Twitter users, such as the Press Complaints Commission and the BBC correspondent Nick Higham, have seen their accounts hijacked after a viral phishing attack [6].

Many researchers along with engineers from Twitter have devoted themselves to keep Twitter as a spam-free online community. They have attempted to protect legitimate users from useless advertisements, pornographic messages or links to phishing or malicious websites. For example, Twitter has published their definitions of spam accounts and The Twitter Rules [8] to protect its users from spam and abuse. Any account engaging in the abnormal activities is subject to temporary or even permanent suspension by Twitter. Meanwhile, many existing research studies, such as [9], [10], [11], [12], [13], also utilize machine learning techniques to detect Twitter spammers.

"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 attackers and defenders leads Twitter spammers to evolve or utilize tools to evade existing detection features [14]. For example, Twitter spammers can evade existing detection features by purchasing followers [15] or using tools to automatically post tweets with the same meaning but different words [16]. This phenomenon has also been observed and discussed by other researchers. For example, Song et. al., observed that spammers tend to use different words with similar semantic meanings to evade detection [17]. Such a phenomenon motivates us to design new and more robust features to detect Twitter spammers.

In this paper, we plan to design more robust features to detect more Twitter spammers through in-depth analysis of the evasion tactics utilized by current Twitter spammers. To achieve our research goals, we collect and analyze around 500,000 Twitter accounts and more than 14 million tweets using Twitter API [18], and identify around 2,000 Twitter spammers by using the blacklist and honeypot techniques. Then, we describe and validate current evasion tactics by both showing case studies and examining four existing state-of-the-art approaches on our collected dataset. Based on the in-depth analysis of those evasion tactics, we design ten new features including graph-based features, neighbor-based features, timing-based features, and automation-based features to detect Twitter spammers. Through our evaluation experiments, we show that our newly designed features can be effectively used

Chao Yang, Robert Harkreader and Guofei Gu are with the Department of Computer Science and Engineering, Texas A&M University, College Station, TX, 77840. E-mail: {yangchao,bharkreader,guofei}@cse.tamu.edu

A preliminary (short) version of this paper appeared in RAID'11 [1]. This material is based upon work supported in part by the National Science Foundation under Grant CNS-0954096.

to detect Twitter spammers. In addition, to better 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 as well as our proposed ones.

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 four state-of-the-art solutions on our collected dataset. Even the best detector still misses detecting around a quarter of Twitter spammers and the worst detector could miss about 40% of the spammers.
- Based on our empirical analysis of Twitter spammers' malicious goals and evasion tactics, we propose and test our newly designed 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%, much higher than that of existing detectors.
- We provide a new framework to formalize the robustness of 24 detection features that are utilized by the existing work and our work, and categorize them into 16 low-robustness features, 4 medium-robustness features and 4 high-robustness features.

II. 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.* [19] conducted 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.* [9] designed diverse metrics to measure Twitter accounts.

Since spam and attacks are so rampant on online social networking sites, Koutrika *et al.* [20] proposed 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.* [21], [22] utilized machine learning techniques to identify video spammers in video social networks such as YouTube. Gao *et al.* [23] presented a study on detecting and characterizing social spam campaigns.

In terms of Twitter, there have 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 [10], [11], [12], [13], 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.* [10] deployed a social honeypot for harvesting deceptive spam profiles from Twitter and also utilized machine learning techniques 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.* [11] utilized 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 [12] designed features such as *reputation score*, *number of duplicate Tweets* and *number of URLs*. The second category of work, e.g. [24], 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/domain blacklists. Especially, to collect training data, both [10] and [13] utilized social honey accounts to attract Twitter spammers.

A recent study "Poultry Markets" [25] focuses on the analysis of those Twitter Account Markets, which could be used by spammers to increase their followers. This work essentially validates the phenomenon that spammers could use tricks to evade existing profile-based features. "Twitter Games" [26] focuses on analyzing how successful spammers pick targets to survive longer in Twitter by analyzing the relationships between spammers' social behaviors and their followers' social behaviors. Thomas *et al.* made a deep analysis of those suspended accounts [27]. Irani *et al.* proposed a user study on reverse social engineering attacks in social networks [28]. Chu *et al.* provided an approach on detecting spam campaigns that manipulate multiple accounts to spread spam on Twitter [29].

Different from existing related work, motivated by our analysis that spammers have evolved to evade existing detection features by using different evasion techniques, our work focuses on designing more robust features with the considerations of both profile-based and content-based feature evasion techniques to detect those evasive spammers. In addition, to better understand the robustness of each detection feature, we also provide a new framework to qualitatively analyze the robustness of detection features. Thus, our work is a valuable supplement to existing Twitter spam detection research.

III. Overview

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

A. Problem Statement

As shown in Section II, most of existing studies on detecting Twitter spam accounts rely on machine learning techniques by designing detection features. However, the arms race nature between the attackers and defenders leads Twitter spammers to evolve or utilize tools to evade existing detection features [14]. 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.

B. Data Collection

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

To crawl Twitter profiles, we have developed a Twitter crawler that taps into Twitter's Streaming API [18]. In order to

decrease the effect of possible sampling bias [30], our crawler recursively collects Twitter accounts in multiple rounds, with the consideration of guaranteeing sampling randomness and maintaining social relationships, rather than simply using the Breath First Search (BFS) sampling technique. More specifically, in each round, our crawler first collects 20 seed Twitter accounts from the public timeline, which are randomly selected by Twitter [31]. Then, the crawler will collect all of those seed accounts' followers and followings. This crawling process will be repeated in the next round. Also, for each account, our crawler collects its 40 most recent Tweets and the URLs in the tweets. 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 million URLs (see Table I).

TABLE I
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

In our work, we use a relatively strict strategy to collect Twitter spammers. More specifically, we focus on those Twitter spammers who post URLs linking to malicious content with an intention to compromise other users' computers or privacy, as mentioned in The Twitter Rules [8]. We target this type of spam accounts due to its prevalence on Twitter and the hazard it poses. Thus, unlike other related work (e.g., [10]), we do not necessarily consider advertisers in Twitter as spammers, unless they post malicious content. To label Twitter spam accounts, we first utilize two methods to detect malicious or phishing URLs in the tweets: *Google Safe Browsing (GSB) [32]* and *URL honeypot*. GSB is a widely used and trustable blacklist to identify malicious/phishing URLs, which is fast but may miss labeling some malicious links. Thus, we also build a high-interaction client-side honeypot based on Capture-HPC [33], which will be used to visit the URL using a real browser in a virtual machine. The honeypot detects a link as malicious, if the visit of the website can lead to the creation/modification of sensitive data (e.g., process, files and registries) in the virtual machine. We define a Tweet that contains at least one malicious or phishing URL as a *Spam Tweet*. In this way, we have collected 3,051 accounts that post at least one Spam Tweet by using GSB and 9,634 such accounts by using honeypot. 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. In this way, we have extracted 2,933 Twitter accounts with a spam ratio higher than 10%. In order to further decrease false positives, our group members spent several days on manually verifying those 2,933 accounts by viewing whether their tweets are useful and meaningful. Finally, we have obtained 2,060 identified spam accounts.

We admit that our data collection strategy might still incur some sampling bias. Also, due to practical limitations, we could only sample a portion of the whole Twittersphere. Thus, our dataset might not contain complete neighbor accounts' information for some collected accounts. However, it is a common challenge for all such line of work to collect a perfect and large-scale real-world OSN dataset with complete social relationships. Also, we believe that our major conclusions obtained by using our sample dataset could still hold, even though the values of some metrics measured (e.g., Graph-based Features proposed in Section V) in this study may vary by using different datasets. We will further discuss the limitations in Section VIII.

After collecting the data, we first make an in-depth analysis of the evasion tactics utilized by spammers through reproducing four state-of-the-art detection schemes and analyzing those missed spammers (false negatives) in Section IV. Then, according to those analysis, we design new and robust detection features to counter these tactics in Section V and formalize the robustness of the detection features in Section VI. Finally, in Section VII, we show that our newly designed features can be an effective supplement to the existing detection approaches.

IV. ANALYZING EVASION TACTICS

This section will describe the evasion tactics utilized by Twitter spammers to evade existing features for spammer detection. Then, we validate these tactics by both showing some case studies and examining this scenario on four state-of-the-art detection approaches based on our collected dataset.

A. Description of Evasion Tactics

The evasion tactics utilized by Twitter spammers can be mainly categorized into the following two types: profile-based feature evasion tactics and content-based feature evasion tactics.

1) Profile-based Feature Evasion: A common intuition for discovering Twitter spam accounts can originate from accounts' basic profile information such as the number of followers and the number of tweets, because 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* and *reputation score* [12]. *Fofo ratio* is the ratio of the number of an account's followings to its followers, which are widely used in existing approaches [10], [13]. Reputation score, which is the ratio of the number of an account's followers to the sum of its followers and followings, could be viewed as a variant of *Fofo ratio*. To evade these features or break Twitter's 2,000 Following Limit Policy[1] [34], spammers can mainly

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

adopt the following strategies to gain more followers. The first strategy is to purchase followers from some third-party websites, which 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 some third-party websites, which use existing customers' accounts to follow new customers' accounts. Since this method only requires 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 to follow their spam 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. [13]). To evade this feature, spammers can post more Tweets to behave more like legitimate accounts, especially by utilizing particular public tweeting tools or software [35].

2) Content-based Feature Evasion: Another common indicator of spam accounts is the content of a suspect account's Tweets. As discussed in Section I, a lot of spam accounts make profits by luring 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 [10], [13], [12]. 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 automated tweeting tools, their published tweets show strong homogeneous characteristics. In this way, many existing approaches design content-based features such as *tweet similarity* [10], [13] and *duplicate tweet count* [12] to detect spam accounts. To evade such content-based detection features, spammers mainly utilize the tactics of 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*, *hashtag ratio* [10], [12]. These normal tweets without malicious URLs may be hand-crafted or obtained from arbitrary users' tweets or consist 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 content-based features such as *tweet similarity* and *duplicate tweet count*. Specifically, 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 [16], can help spammers to spin spam tweets into hundreds of variable tweets with the same semantic meaning but different words.

B. Validation of Evasion Tactics

In this section, we aim at validating those evasion tactics described in the previous section by first showing real case studies and public services/tools that can be utilized by the spammers. Then, to further validate such tactics on our collected dataset, we also reproduce existing detection schemes [10], [13], [12], [11] and evaluate them on our collected examination dataset. (To better explain our result, we label the work of [10] as A, [13] as B, [12] as C, and [11] as D.) Rather than accurately finding out all evasive accounts in our dataset (because it is difficult or even impossible to obtain such a ground truth on a large scale dataset), we focus on analyzing those missed spammers (false negatives) in the existing approaches. Through analyzing the reason why those spammers are missed by existing approaches, we validate that those spammers have indeed evolved to behave similar as legitimate accounts to evade existing detection features.

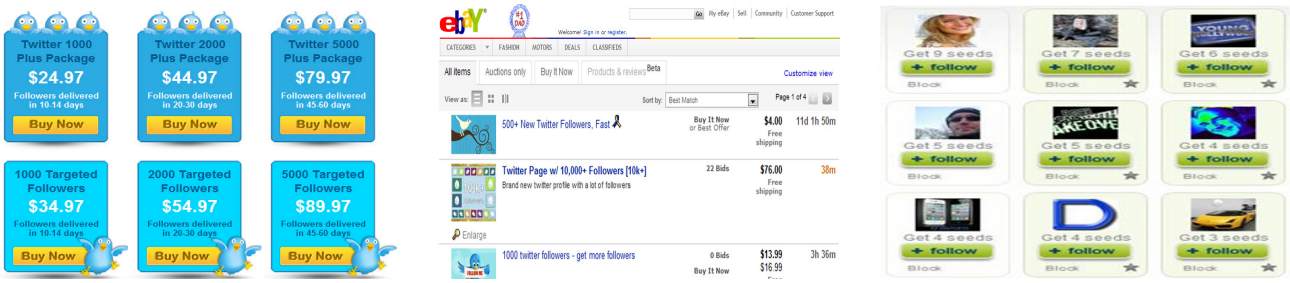
Gaining More Followers: As described in Section IV-A, spammers can gain more followers by purchasing them, exchanging them and creating fake accounts. In fact, several public websites allow for direct purchase of followers, even though the rates per follower in each website may vary, as seen in Table II.

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

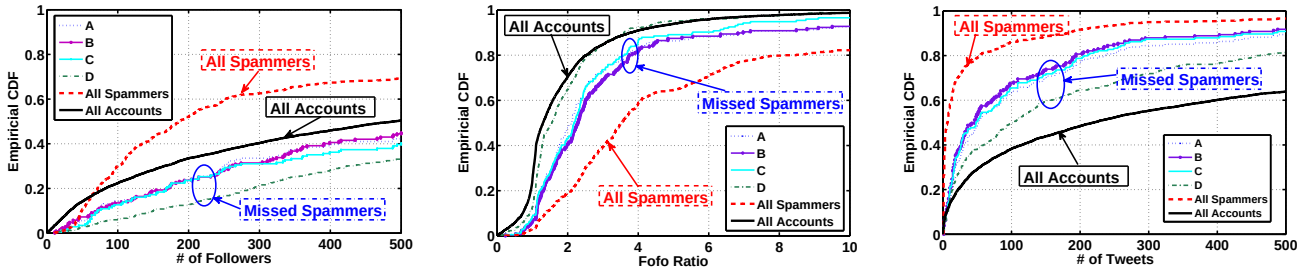
Also, Fig. 1(a) shows a real online website, from which users can directly purchase followers at a very cheap price. The website also claims that users can buy targeted followers with specific keywords in their tweets at a much higher price. As seen in Fig. 1(b), Twitter followers can even be directly purchased through the famous and widely used online bidding website – eBay. Besides purchasing followers, Twitter spammers can also increase their followers by exchanging followers with other users. Fig. 1(c) shows a real online website from which users can increase their followers through obtaining seeds by following other accounts.

After showing these online services through which spammers can obtain more followers, we examine two widely used detection features of *number of followers* and *fofo ratio* on those four existing work [10], [13], [12], [11] based on our collected dataset. In particular, we draw the distributions of these two metrics of three sets of accounts: missed spammers (false negatives) in each of four existing approaches, *all accounts* (around $500K$ collected accounts), and *all spammers* (2,060



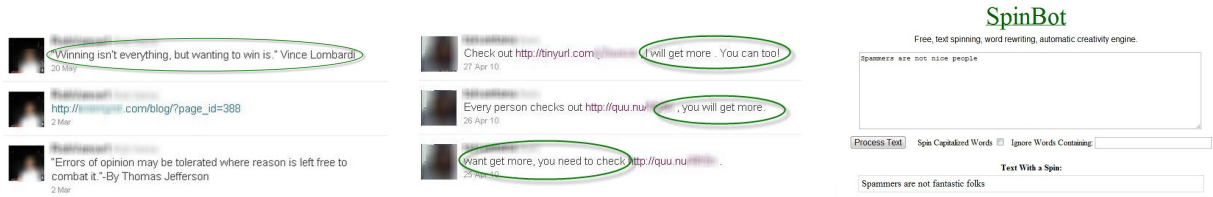
(a) Purchasing Followers (b) Bidding Followers (c) Exchanging followers

Fig. 1. Online Twitter Follower Trading Website



(a) Number of Followers (b) Fofo Ratio (c) Number of Tweets

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



(a) Mixing Normal Tweets (b) Posting Heterogeneous Tweets (c) Spin Bot

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

identified spammers). 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 Web to post tweets, spammers can utilize tools such as AutoTwitter [35] and Twitter API [18] to automatically post more tweets. Fig. 2(c) shows the distributions 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 scam websites.

Posting Heterogeneous Tweets: To avoid detection features such as *tweet similarity* and *duplicate tweet count*, spammers could use tools to post heterogeneous tweets with the same semantic meaning but with 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". One example of such tools that can be used to automatically create 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 yet different words is generated in the small text field below.

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. After analyzing these evasive tactics utilized by spammers, we next design new features as a complement to existing ones, which make spammers more difficult to evade detection.

V. 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 significant 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.

More specifically, graph-based features and neighbor-based features are mainly designed to catch those spammers who attempt to evade profile-based features by controlling their own social behaviors (e.g., increasing the number of followers or tweets). The intuition is that even though spammers could change their own social behaviors, it is still very difficult for them to change the the social behaviors of the majority of their (benign) following/follower accounts, i.e., spammers typically could not force their followings or followers to follow specific accounts or to post specific tweets. Thus, the values of those graph-based and neighbor-based features could be used to distinguish spammers from normal accounts. In addition, automation-based and timing-based features are mainly designed to catch those spammers who attempt to evade content-based features by adding more benign tweets. The intuition is that even though spammers could insert normal tweets, they typically still need to use customized tools or software to automatically post a considerable number of malicious tweets with malicious URLs to trap victims more effectively. Thus, those automated social behaviors of spam accounts could still indicate the differences between themselves and normal accounts.

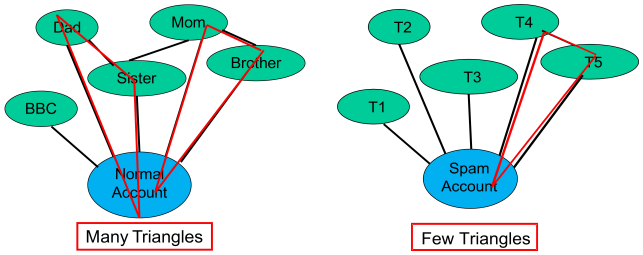
A. Graph-based Features

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 of 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 [36]. 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 with 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 coefficients. This intuition can be illustrated in Fig. 4(a) and (b). Compared with spam accounts, the stronger social relationships among normal accounts intend to will form more triangles.



(a) Normal Account (b) Spam Account

Fig. 4. Illustration of the differences of local clustering coefficient between normal accounts and spam accounts.

In our implementation, to calculate the local clustering coefficient for each account, we first collect its followers as neighbors. Thus, we could build a small social graph based on this account and its followers. Then, its local clustering coefficient could be calculated based on examining how complete this small social graph is.

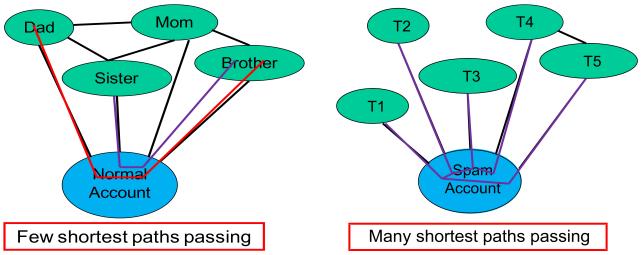
Betweenness Centrality is a centrality measure of a vertex within a graph [37]. Vertices that occur on many shortest paths between other vertices have a higher betweenness than those that do not. In a directed graph, betweeness centrality of each vertex v can be computed with Eq. (2), where δ_{st} is 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 randomly follow many unrelated accounts. As a result, when the Twitter spammer follows these unrelated accounts, the spammer creates a new shortest path between those accounts through the spam account. Thus, the betweenness centrality of the spammer will be high. This intuition can be illustrated in Fig. 5(a) and (b). Compared with normal accounts, spammer accounts' randomly following policy essentially create more shortest paths between their following accounts passing through them.

An informed spammer would be able to carefully choose the accounts to follow 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, skills and money to implement, but also limits the number of potential victims of spammers.

Bi-directional Links Ratio: If two accounts follow each other, we consider there is a bidirectional link between them. The number of bi-directional links of an account reflects



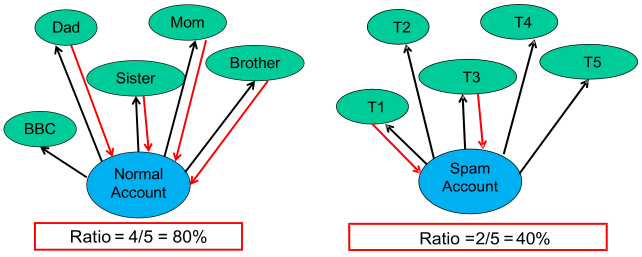
(a) Normal Account (b) Spam Account

Fig. 5. Illustration of the differences of betweeness centrality between normal accounts and spam accounts.

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/her 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 with 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.



(a) Normal Account (b) Spam Account

Fig. 6. Illustration of the differences of bi-directional link ratios between normal accounts and spam accounts.

The intuition behind this feature can be illustrated in Fig. 6(a) and (b). Since it is very difficult for spammers to force their following accounts to follow them back, compared with their high values of N_{fing}, their values of R_{bilink} will be relatively difficult to increase. Thus, compared with normal accounts, spammers intend to have much smaller values of bi-directional link ratios. To validate such an intuition, we compare the values of *bi-directional links ratio* in all our collected accounts and our identified spam accounts. As seen in Figure 7, spammer accounts tend to have smaller values of bi-directional links ratio. In particular, around 70% of spammers' values are less than 0.2, while only around 50% of all accounts' values are less than 0.2.

We acknowledge that spammers could try to change the values of these graph features by building fake social relationships with other spam accounts. However, this strategy could still be difficult to significantly impact the global graph features. In addition, by doing this kind of evasion, the efficiency of

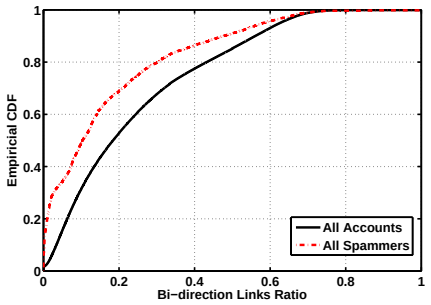


Fig. 7. The comparison of values of bi-directional link ratios between all accounts and spam accounts.

spammers to trap victims will be greatly reduced (see more information in Section VI). Furthermore, if spammer accounts tend to follow with each other, they could be identified by existing sybil-attack detection approaches (e.g., SybilGuard [38]) or relationship analysis approaches (e.g., [39]).

B. 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 account's 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 carefully selected accounts that typically have higher quality unlike the spammers typically do blind/random following. 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). Thus, we also design another more 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 as 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 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, leading 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 that 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 more difficult to be evaded.

C. Automation-based Features

Due to the high 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[2] Ratio, API URL Ratio and API Tweet Similarity.

API Ratio is the ratio of the number of tweets with the tweet source as "API" to the total number of tweets. As existing work [41] shows, many bots use API to post tweets, so a higher API ratio implies this account is more suspicious.

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. 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 IV. 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.

API Tweet Similarity: Spammers can use tricks to evade the detection feature of *tweet similarity* as described in Section IV 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.

[2]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" [40]. Thus, in this paper, we use "API" to refer those unregistered third-party applications.

D. Timing-based Features

Similar to other timing-based features such as *tweeting rate* presented in [11], we design another feature named *following rate*.

Following Rate reflects the speed at which an account follows other accounts. Since spammers 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 constraints. We analyze its robustness in Section VI, and discuss its limitation in Section VIII.

After designing these new features, we first formalize the robustness of most of the existing detection features and our designed features in Section VI. 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 VII.

VI. Formalizing Feature Robustness

In this section, in order to analyze how to design effective features to detect Twitter spammers along with their evolutions, we formalize the robustness of those detection features.

A. Formalizing the Robustness

Before analyzing the robustness of each feature in detail, we first build a model to define the robustness of the detection features. Due to the dual-purpose nature of 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 with 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 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. And 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

TABLE III
ROBUSTNESS OF DETECTION FEATURES

Index	Category	Feature	Used in Work	Robustness
F_1	Profile	number of followers (N_{fer})	C, D	Low
F_2 (+)	Profile	number of followings (N_{fing})	B, C, D, ours	Low
F_3 (+)	Profile	fofo ratio (R_{fofo})	A, B, D, ours	Medium
F_4	Profile	reputation (Rep)	C	Medium
F_5 (+)	Profile	the number of tweets (N_{twt})	B, D, ours	Low
F_6 (+)	Profile	age	A, ours	High
F_7 (+)	Content	URL ratio (R_{URL})	A, B, C, D, ours	Low
F_8 (+)	Content	unique URL ratio	A, ours	Low
F_9	Content	hashtag(#) ratio	C, D	Low
F_{10}	Content	reply(@) or retweet ratio	A, C, D	Low
F_{11} (+)	Content	tweet similarity (T_{sim})	A, B, ours	Low
F_{12}	Content	duplicate tweet count	C	Low
F_{13}	Content	spam word ratio	D	Low
F_{14}	Graph	number of bi-directional links (N_{bilink})	A	Low
F_{15} (*)	Graph	bi-directional links ratio (R_{bilink})	ours	Medium
F_{16} (*)	Graph	betweenness centrality (BC)	ours	High
F_{17} (*)	Graph	clustering coefficient (CC)	ours	High
F_{18} (*)	Neighbor	average neighbors' followers (A_{nfer})	ours	Low
F_{19} (*)	Neighbor	average neighbors' tweets (A_{ntwt})	ours	Low
F_{20} (*)	Neighbor	followings to median neighbors' followers (R_{fing_mnfer})	ours	High
F_{21} (*)	Timing	following rate (FR)	ours	Low
F_{22} (+)	Timing	tweet rate (TR)	A, D, ours	Low
F_{23} (*)	Automation	API ratio (R_{API})	ours	Medium
F_{24} (*)	Automation	API URL ratio (R_{API_URL})	ours	Medium
F_{25} (*)	Automation	API Tweet Similarity (T_{api_sim})	ours	Medium

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 is shown in Table III[3]. (Similar to Section IV, we also label the work of [10] as A, [13] as B, [12] as C, and [11] as D.)

B. Robustness of Profile-based Features

As described in Section IV, 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 with Eq.(7).

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

From Table II, we can find that C_{fer} can be very cheap, which could be much smaller compared with the P_{fing}. Even when the spammers who desire to follow 2,000 accounts, which is restricted by the Twitter's 2,000 Following Limit Policy [34], they just need to spend $50 to bypass 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. (Obviously, it is also possible to evade if the spammers can use some tricks to obtain the Twitter accounts that were registered in Twitter a long time ago. However, 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 [42].)

Since *number of tweets* (F_5) is related to several content-based features, we show the analysis of (F_5) shortly.

C. Robustness of Content-based Features

As shown in Table III, content-based features can be divided into two types: signature-based features (F_7, F_8, F_9, F_9 and F_{13}) 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 IV, 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 generality, we analyze robustness of the *URL_ratio* (F_7) to represent the analysis of this type of features. Similar to Eq.(7), if a spammer needs to post N_{mt} tweets with malicious URLs, the robustness for F_7 can be computed with Eq.(8).

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

From Eq.(8), we can find that if spammers utilize software such as AutoTwitter [35] and Twitter API [18] 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), compared with the big profits of successfully alluring the victims to click the malicious URLs, the cost is still small.

[3]Some features used in different existing work are designed with the same intuition but implemented in slightly different ways. For example, in terms of the feature – "URL ratio", in the work D, it computes four values of this feature for each tweet including the maximum, minimum, average, and median. Thus, in this table, we only list main features designed based on different intuitions rather than listing all derivative features

D. Robustness of Graph-based Features

For the graph-based features, we can divide them into two types: *reciprocity-based features* (F_{14} and F_{15}) based on the number of the bi-directional links and *position-based features* (F_{16} and F_{17}) 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_{14} and F_{15} can be computed with Eq. (9) and (10). Since attackers could not achieve obvious profit (e.g. sending more tweets or obtaining more followers to garner victims) by obtaining more bi-directional links to evade reciprocity-based features, we mainly consider their cost rather than profit.

$$R(F_{14}) = T_{F_{14}} \cdot C_{BiLink} \tag{9}$$

$$R(F_{15}) = T_{F_{15}} \cdot N_{fing} \cdot C_{BiLink} \tag{10}$$

Since it is impractical to set a high bi-directional link threshold to distinguish legitimate accounts and spammers, the value of $T_{F_{14}}$ could not be set high. Meanwhile, when $T_{F_{14}}$ is small, spammers can obtain bi-directional links by following back their followers. Thus, the C_{BiLink} is neither high. Thus, from Eq. 9, we can find that the value of $R(F_{14})$ is not high. In terms of feature F_{15}, since the average of the bi-directional links ratio is 22.1% [19] and spammers usually have a large value of N_{fing}, spammers need to obtain much more bidirectional links to show a normal bi-directional links ratio. Even though spammers could try to increase C_{BiLink} by following back their followers, due to the big number of their following accounts and the difficulties of forcing those accounts to follow spammers back, it will cost a lot for spammers 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 hardly control the benign accounts they have 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.

E. 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 V. If we use N_{follow} to denote the number of popular accounts (the accounts that have very big follower numbers) that a spammer needs to follow to get a high enough A_{nfer} to evade feature F_{18}, then the robustness of F_{18} can be computed with Eq.(11).

$$R(F_{18}) = N_{follow} \cdot C_{follow} \tag{11}$$

Since there are many popular accounts with a lot of 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_{19}.

However, for feature F_{20}, 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_{20} is relatively difficult to evade.

F. 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, the cost can still be low because these two features can be totally controlled by the spammers. Thus, feature F_{21} and F_{22} are relatively easy to evade.

G. Robustness of Automation-based Features

As discussed in Section V, in order to more efficiently achieve the malicious goals, many Twitter spammers will control multiple spam accounts to spread the spam tweets. Similar to the discussion of 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, most 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_{23} can be computed with Eq. (12).

$$R(F_{23}) = N_{spam} \cdot [\frac{N_{mt}}{T_{F_{23}}} \cdot (1 - T_{F_{23}}) \cdot C_{twt\_web} + N_{mt} \cdot C_{twt\_api}] - N_{spam} \cdot N_{mt} \cdot P_{mt} \tag{12}$$

Since few legitimate accounts would use API to post tweets, $T_{F_{23}}$ 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.

Only using feature F_{23} will bring some false positives, because legitimate accounts may also use API to post tweets. However, if we combine the features of F_{23}, F_{24}, and F_{25} together, it will decrease those false positives, because 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 III.

Note that the model presented above serves as the first attempt to analyze the robustness of detection features. Although more statistically quantitative analysis based on the usage of real-world data could make the model to be more persuasive, it essentially requires the answers of many research questions. For example, how to accurately and quantitatively measure OSN spammers economic profit? How to measure the cost of losing an effective spam account for spammers? What specific behaviors (accurate values in Twitters suspension rules)

will make accounts suspended by Twitter? To answer such questions, more research efforts of analyzing OSN spammers' economic chain and OSNs anti-spam measures are needed, which are out of the scope of this study. We hope this study could stimulate more future research in this direction.

VII. Evaluation

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

We evaluate the feature set by implementing machine learning techniques on two different datasets: Dataset I and Dataset II. Dataset I consists of 20,000 accounts without any spam tweets, which are randomly selected from our crawled $500K$ accounts described in Section III-B, and all 2,060 identified spammer accounts. To decrease the effects of sampling bias and show the quality of our detection feature schema without using URL analysis as the ground truth, we also crawled another 35,000 Twitter accounts and randomly selected 3,500 accounts to build another dataset, denoted as Dataset II.

A. Evaluation on Dataset I

In this section, based on Dataset I, we evaluate our machine learning feature set including *performance comparison*, *feature validation*, *learning curve*, *Feature Rank*, and *Varying Spam Account Prior Probability*.

Performance Comparison: In this experiment, we compare the performance of our work with four existing approaches[4]: [10] using 10 features, [13] using 6 features, [12] using 7 features and [11] using 62 features. (Similar as before, to better show the results, we label the work [10] as A, [13] as B, [12] as C, [11] as D, and our work as E.) We conduct our evaluation by using four different machine learning classifiers: *Random Forest (RF)*, *Decision Tree (DT)* , *Bayes Net (BN)* and *Decorate (DE)*. For each machine learning classifier, we use the method of *ten-fold cross validation* to compute three performance metrics: *False Positive Rate*, *Detection Rate*, and *F-1 measure.*[5]

As seen in Fig. 8, our approach outperforms all existing work. Specifically, from Fig. 8(a), we can find that the false positive rates of our work under all four machine learning classifiers are the lowest and can be steadily maintained under 1%. Especially, under the Random Forest classifier (RF), the false positive rate of our work is only 0.4%, while this rate is 0.6% for the best detector in existing work (RF of D) and 1% for the second best detector (RF of A). From Fig. 8(b), we can see that the detection rates of our work under all four machine learning classifiers are also the highest. In particular, the highest detection rate of our work among these four classifiers is 85.4%, and the lowest detection rate of our work is around 84%. As a comparison, the detection rate is only 61% for the worst detector in existing work (DE in C), and 78% for the best existing detector (RF in D), although D utilizes significantly more features than us (over 60 features used in D). Fig. 8(c) also 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 [10], [13] and [11] is better than that of [12]. That is mainly because all these three studies utilize content-based feature such as *tweet similarity* and *spam word ratio*. However, [12] only uses the feature of *duplicate tweet count*, which may has been widely evaded by spammers. Also, [11] utilizes various timing-based and content-based features leading its performance to be better than that of [10] and [13].

Feature Validation: To further validate that the improved performance results of our work is indeed due to our designed features rather than the simple combination of multiple existing features, we also implement another experiment to compare the performance of two feature sets – without and with using our new features. Specifically, the first one consists of 8 existing features used in the previous experiment. The second one further adds our new features. Table VI shows that for each classifier, with the addition of our newly designed features, the detection rate (DR) increases over 10%, while maintaining an even lower false positive rate (FPR). This observation validates that the improvement of the detection performance is indeed due to our newly designed features.

TABLE IV
Comparison Without and With New Features

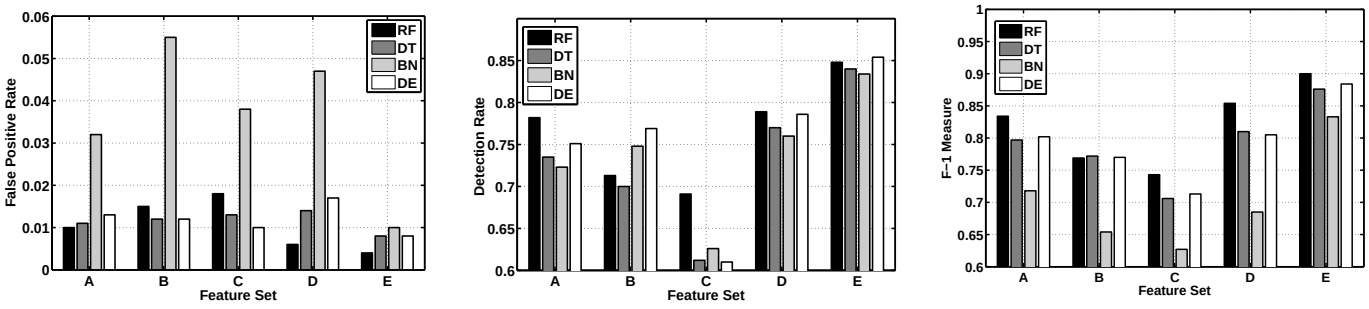
	Without Our Features			With Our Features		
Classifier	FPR	DR	F-1 Measure	FPR	DR	F-1 Measure
Random Forest	0.013	0.737	0.791	0.004	0.848	0.9
Decision Tree	0.014	0.697	0.760	0.008	0.840	0.876
BayesNet	0.068	0.762	0.629	0.01	0.838	0.833
Decorate	0.012	0.697	0.768	0.012	0.854	0.884

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 evaluate the performance of *False Positive Rate*, *Detection Rate*, and *Accuracy* for our work using five different training ratios (1:1, 3:1, 5:1, 7:1, 9:1). As Fig. 9(a) shows, even when the training ratio varies, our approach can obtain a low false positive rate, never rising above 0.5%. Fig. 9(b) shows that our approach can steadily detect over 82% spammers under different training ratios. From Fig. 9(c), we can see that the total accuracy can remain higher than 98%, even when the training ratio is small (e.g. 1:1).

Feature Rank: To show the effectiveness of the features in our feature set, we use three prevalent feature evaluation methods: Information Gain, Chi-Square test and AUC (Area Under the ROC Curve). For each method, we compute the ranks of the detection features that we used (seen in Table V). The results show that our new designed features are effective to detect Twitter spam accounts. In particular, our designed automation-based features (API ratio and API URL ratio) are highly ranked in all three different methods. In addition, our newly designed graph-based features (e.g., average neighbors' tweets) are also very effective in discovering spammers. It is worth noting that the ranks of features in this table do not

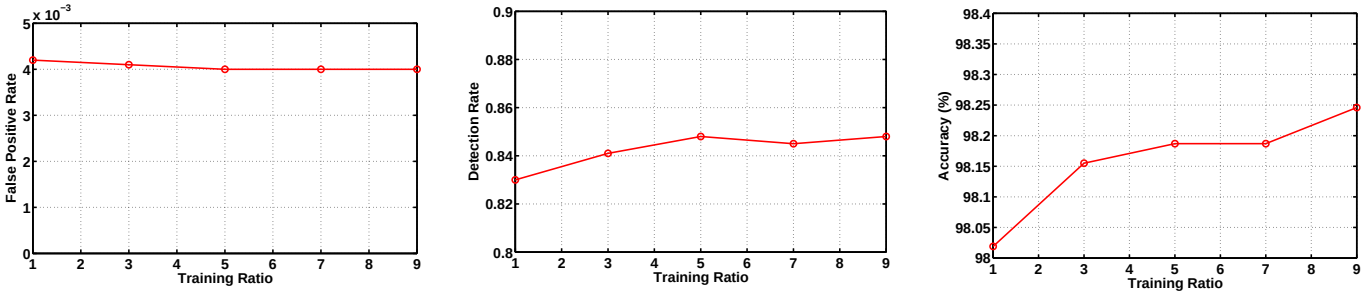
[4]The features used in those approaches can be seen in Table III.

[5]F-1 measure [43] is a measure with the consideration of both precision and recall.



(a) False Positive Rate (b) Detection Rate (c) F-1 Measure

Fig. 8. Performance comparison with the existing approaches



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

Fig. 9. Performance comparison on different training ratios

have any relationships to their robustness. That is, although some features (e.g., fofo ratio) are still effective to distinguish most current spammers (i.e., they still have good ranks in the table), they could be easily evaded by spammers, leading to low robustness. On the other hand, although some of our new features do not necessarily have very high ranks in the table, they can contribute to the detection of very evasive spammer accounts, as already demonstrated previously.

Varying Spam Account Prior Probability: In this experiment, we also evaluate the performance of our work on different datasets with different spam account prior probabilities. Specifically, for each dataset, we use the same 2,000 identified spam accounts which are randomly sampled from Dataset I. Then, we adjust the ratio of the number of spam to normal accounts on different datasets to tailor the performance, where a larger ratio indicates a stronger tendency to classify a account as a spam account. We evaluate the performance of *False Positive Rate*, *Detection Rate*, and *Accuracy* on four different datasets with four ratios (1:1, 1:2, 1:5, 1:10). The result can be seen in Table VI. This result shows that our approach could achieve a relatively high accuracy score (F-1 measure higher than 0.89) in all four different datasets.

TABLE VI
COMPARISON BY VARYING SPAM ACCOUNT PRIOR PROBABILITY

Ratio	FPR	DR	F-1 Measure
1:1	0.053	0.947	0.947
1:2	0.025	0.921	0.934
1:5	0.009	0.879	0.913
1:10	0.003	0.853	0.894

B. 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 new dataset containing 3,500 unclassified Twitter accounts, which are randomly selected from Twitter. 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. Due to the lack of the ground truth on this dataset, we do not know the exact number of spam accounts. Thus, we evaluate our approach by computing the Bayesian detection rate [44] – the probability of actually being a spammer, whenever an account is reported as a spammer account by the detection system.

Specifically, we use Dataset I as the training dataset, and Dataset II as the testing data. Then, based on our detection feature set, we use BayesNet classifier to predict spammers on Dataset II. Finally, the classifier reported 70 spammers. Through the manual investigation of those 70 spammers, we found 37 accounts post spam and 25 accounts are suspicious promotional advertisers. In this case, we have a high Bayesian detection rate of 88.6% (62/70). Then, we further investigate the other 8 false positives. We find that all of them have odd behaviors, but have not posted malicious content. Specifically, 6 of them are actively and repeatedly tweeting the same topic. The other 2 have posted very few tweets, yet have a large number of followings with a high ratio of followings to followers.

VIII. Limitation and Future Work

Due to practical constraints, we can only crawl a portion of the whole Twittersphere and our crawled dataset may still have sampling bias. However, collecting an ideal large dataset

TABLE V
FEATURE RANK

Index	Feature	Information Gain (Rank)	Chi-Square (Rank)	AUC (Rank)
F_2	number of followings	12	10	8
F_3	fofo ratio	1	9	15
F_5	number of tweets	2	7	10
F_6	account age	1	8	7
F_7	URL ratio	4	3	2
F_8	unique URL ratio	5	5	1
F_{11}	tweet similarity	2	12	9
F_{15}	bi-directional links ratio	6	15	12
F_{16}	betweenness centrality	3	18	17
F_{17}	clustering coefficient	7	14	6
F_{18}	average neighbors' followers	9	16	13
F_{19}	average neighbors' tweets	10	4	5
F_{20}	followings to median neighbors' followers	17	17	16
F_{21}	following rate	14	11	18
F_{22}	tweet rate	4	6	11
F_{23}	API ratio	1	1	3
F_{24}	API URL ratio	2	5	4
F_{25}	API tweet similarity	15	13	14

from Twitter, a large and dynamic real-world OSN, without any bias is almost a mission impossible.

In addition, it is well acknowledged in the community that it is challenging (or impossible) to achieve a comprehensive ground truth for Twitter spammers. Also, in order to guarantee that our collected spammers are real spammers, we use a more strict strategy than what used in most of other related work to collect our spammers. Thus, the number of our identified spammers is only a lower bound, and the percentage of identified spammers in our dataset may be smaller than that reported in other studies. However, even for a subset of spammers, we can see that they are evolving to evade detection. And our evaluation validates the effectiveness of our newly designed features to detect these evasive 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, it is meaningful to detect them.

We clearly admit that those 20K accounts used as our benign dataset may still contain some spam accounts. However, it is very difficult to obtain a perfect ground truth from such a big dataset. Thus, we only collect those accounts without posting malicious URLs to build the benign dataset. Also, we believe that our major conclusion could still be held, although there could be some noisy items in the training dataset.

While graph-based features such as local clustering coefficient and betweenness centrality are relatively difficult to evade, these features are also expensive to extract. Also, precisely calculating the values of such graph metrics on large graphs (e.g., the whole Twitter graph) is very challenging and a hot research issue, which is out of scope of this work. However, we could still estimate the values of these two features by using a neighbor-sampling technique that allows us to compute these metrics piece-by-piece. 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, we plan to design more robust features, evaluate our machine learning detection scheme on larger datasets by using more crawling strategies, 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. In addition, further studies on analyzing the correlation among different features and designing better machine learning classifiers by selecting more effective features are also in our future plan.

IX. 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 four 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 is also much higher than all existing detectors under four different prevalent machine learning classifiers. To promote further research in this area, we have also released some of our datasets.[6]

REFERENCES

[1] C. Yang, R. Harkreader, and G. Gu, "Die free or live hard? empirical evaluation and new design for fighting evolving twitter spammers," in *Proceedings of the 14th International Symposium on Recent Advances in Intrusion Detection (RAID'11))*, September 2011.

[2] "Costolo: Twitter Now Has 190 Million Users Tweeting 65 Million Times A Day," http://techcrunch.com/2010/06/08/twitter-190-million-users/, 2010.

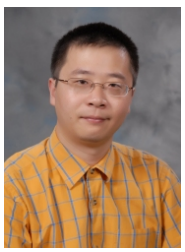
[3] "Acai Berry spammers hack Twitter accounts to spread adverts," http://nakedsecurity.sophos.com/2010/12/13/acai-berry-spam-gawker-password-hack-twitter/, 2009.

[6] http://faculty.cse.tamu.edu/guofei/research_release.html

[4] "New Koobface campaign spreading on Facebook," http://forums.cnet.com/7726-6132_102-5064273.html, 2011.
[5] "Twitter-based Botnet Command Channel," http://ddos.arbornetworks.com/2009/08/twitter-based-botnet-command-channel/, 2009.
[6] "Twitter phishing hack hits BBC, Guardian and cabinet minister." http://www.guardian.co.uk/technology/2010/feb/26/twitter-hack-spread-phishing, 2010.
[7] "A new look at spam by the numbers," http://scitech.blogs.cnn.com/2010/03/26/a-new-look-at-spam-by-the-numbers/, 2010.
[8] "The Twitter Rules," http://help.twitter.com/entries/18311-the-twitter-rules, 2011.
[9] M. Cha, H. Haddadi, F. Benevenuto, and K. P. Gummadi., "Measuring User Influence in Twitter: The Million Follower Fallacy," in *Int'l AAAI Conference on Weblogs and Social Media (ICWSM)*, 2010.
[10] K. Lee, J. Caverlee, and S. Webb., "Uncovering Social Spammers: Social Honeypots + Machine Learning," in *ACM SIGIR Conference (SIGIR)*, 2010.
[11] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida., "Detecting Spammers on Twitter," in *Collaboration, Electronic messaging, Anti-Abuse and Spam Conffference (CEAS)*, 2010.
[12] A. Wang., "Don't follow me: spam detecting in Twitter," in *Int'l Conferene on Security and Cryptography (SECRYPT)*, 2010.
[13] G. Stringhini, S. Barbara, C. Kruegel, and G. Vigna, "Detecting Spammers On Social Networks," in *Annual Computer Security Applications Conference (ACSAC'10)*, 2010.
[14] "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, 2010.
[15] "Purchase followers," http://http://buyafollower.com/, 2011.
[16] "Tweet spinning your way to the top," http://blog.spinbot.com/2011/03/tweet-spinning-your-way-to-the-top/, 2011.
[17] J. Song, S. Lee, and J. Kim, "Spam Filtering in Twitter Using Sender-Receiver Relationship," in *in Proceedings of the 14th International Symposium on Recent Advances in Intrusion Detection (RAID'11)*, 2011.
[18] "Twitter API in Wikipedia," http://apiwiki.twitter.com/, 2011.
[19] 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.
[20] 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.
[21] F. Benevenuto, T. Rodrigues, V. Almeida, J. Almeida, C. Zhang, and K. Ross., "Identifying Video Spammers in Online Social Networks," in *Int'l Workshop on Adversarial Information Retrieval on the Web (AirWeb'08)*, 2008.
[22] F. Benevenuto, T. Rodrigues, V. Almeida, J. Almeida, and M. Gonalves., "Detecting Spammers and Content Promoters in Online Video Social Networks," in *ACM SIGIR Conference (SIGIR)*, 2009.
[23] 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.
[24] 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.
[25] G. Stringhini, M. Egele, C. Kruegel, and G. Vigna, "Poultry Markets: On the Underground Economy of Twitter Followers," in *Proceedings of Workshop on Online Social Networks*, 2012.
[26] V. Sridharan, V. Shankar, and M. Gupta, "Twitter Games: How Successful Spammers Pick Targets." in *Proceedings of 28th ACSAC*, 2012.
[27] K. Thomas, C. Grier, V. Paxson, and D. Song, "Suspended Accounts in Retrospect:An Analysis of Twitter Spam." in *Internet Measurement Conference (IMC'11)*.
[28] D. Irani, M. Balduzzi, D. Balzarotti, E. Kirda, and C. Pu, "Reverse Social Engineering Attacks in Online Social Networks." in *Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA)*, 2011.
[29] Z. Chu, I. Widjaja, and H. Wang, "Detecting Social Spam Campaigns on Twitter." in *10th International Conference of Applied Cryptography and Network Security (ACNS'12)*.
[30] 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.
[31] "Twitter Public Timeline," http://twitter.com/public_timeline.
[32] "Google Safe Browsing API," http://code.google.com/apis/safebrowsing/.
[33] "Capture HPC," https://projects.honeynet.org/capture-hpc.
[34] "The 2000 Following Limit Policy On Twitter," http://twittnotes.com/2009/03/2000-following-limit-on-twitter.html, 2009.
[35] "Auto Twitter," http://www.autotweeter.in/, 2011.
[36] "Local Clustering Coefficient," http://ikipedia.org/wiki/Clustering_coefficient#Local_clustering_coefficienty.
[37] "Betweenness Centrality," http://en.wikipedia.org/wiki/Centrality.
[38] H. Yu, M. Kaminsky, P. Gibbons, and A. Flaxman, "SybilGuard: Defending Against Sybil Attacks via Social Networks." in *Proceedings of ACM SIGCOMM Conference*, 2006.
[39] C. Yang, R. Harkreader, J. Zhang, S. Shin, and G. Gu, "Analyzing spammers' social networks for fun and profit – a case study of cyber criminal ecosystem on twitter," in *Proceedings of the 21st International World Wide Web Conference (WWW'12)*, April 2012.
[40] "TweetDeck." http://www.tweetdeck.com/.
[41] 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.
[42] "Twitter account for sale," http://www.potpiegirl.com/2008/04/buy-sell-twitter-account/, 2008.
[43] "F1 Score." http://en.wikipedia.org/wiki/F1_score.
[44] S. 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*, 1999, pp. 1–7.

Chao Yang is a Ph.D. candidate in the Department of Computer Science and Engineering at Texas A&M University. He received his B.S. degree in Mathematics and M.S. degree in Computer Science from Harbin Institute of Technology in China. His research interests include network security, web security (especially social networking website security) and smartphone security.

Robert Harkreader is a M.S. graduate from Department of Computer Science and Engineering at Texas A&M University. During his Master's career, he was working on security issues in the social network websites with Dr. Guofei Gu in SUCCESS Lab. Upon his graduation, he started working for Cisco.



Guofei Gu is an assistant professor in the Department of Computer Science & Engineering at Texas A&M University (TAMU). Before coming to Texas A&M, he received his Ph.D. degree in Computer Science from the College of Computing, Georgia Institute of Technology. His research interests are in network and system security, such as malware analysis/detection/defense, social web security, cloud and software-defined networking (SDN/OpenFlow) security. Dr. Gu is a recipient of 2010 NSF CAREER Award, 2013 AFOSR Young Investigator Award, and 2010 IEEE Symposium on Security and Privacy (Oakland'10) best student paper award. He is currently directing the SUCCESS (Secure Communication and Computer Systems) Lab at TAMU.