CATS: Characterizing Automation of Twitter Spammers

Amit A. Amleshwaram, Narasimha Reddy, Sandeep Yadav
Department of Electrical and Computer Engineering
Texas A&M University
College Station, TX - 77843
Email: {amitiitg@cse. , reddy@ece. , sandeepy@}tamu.edu

Guofei Gu, Chao Yang
Department of Computer Science and Engineering
Texas A&M University
College Station, TX - 77843
Email: {guofei@, yangchao@}cse.tamu.edu

***Abstract*—Twitter, with its rising popularity as a micro-blogging website, has inevitably attracted the attention of spammers. Spammers use myriad of techniques to evade security mechanisms and post spam messages, which are either unwelcome advertisements for the victim or lure victims in to clicking malicious URLs embedded in spam tweets. In this paper, we propose several novel features capable of distinguishing spam accounts from legitimate accounts. The features analyze the behavioral and content entropy, bait-techniques, and profile vectors characterizing spammers, which are then fed into supervised learning algorithms to generate models for our tool, CATS. Using our system on two real-world Twitter data sets, we observe a 96% detection rate with about 0.8% false positive rate beating state of the art detection approach. Our analysis reveals detection of more than 90% of spammers with less than five tweets and about half of the spammers detected with only a single tweet. Our feature computation has low latency and resource requirement making fast detection feasible. Additionally, we cluster the unknown spammers to identify and understand the prevalent spam campaigns on Twitter.**

I. Introduction

Online social networks have become the preferred form of communication not only between friends and family, but also for business means. Twitter is one such popular network where the short message communication (called *tweets*) has enticed a large number of users. The tweets exchanged between users are 140 character messages which may even be embedded with URLs (with the help of URL shortening services). Twitter's wide reach has also attracted spammers looking to mint financial gains through easy access to millions of users.

Spammers on Twitter employ myriad of techniques to post unwanted tweets to users of an online social network such as Twitter. Such tweets pose either as advertisements, scams and help perpetrate phishing attacks or the spread of malware through the embedded URLs. To gain a wider reach to potential victims, spammers are known to befriend (or to *follow* in Twitter terminology) unrelated users, send unsolicited messages and masquerade malicious components (for instance, using URL shorterners to substitute *malicious-appearing* URLs).

While prohibiting tweets with undesired text, is essential to keep users from being annoyed, preventing spam proliferation also translates to protecting users from clicking malicious links in the tweets. The malicious URLs pose threats in the form of drive-by-downloads [34] and other infections which allows the installed malware to siphon host information. The infected machine may also assist in nefarious botnet activities such as by itself being a source of email spam or used during the execution of Distributed Denial of Service (DDoS) attacks. Additionally, the embedded links may be used to launch phishing attacks where users may be duped into disclosing confidential information. From Twitter's perspective, spam threatens to prohibit the growth of user base hurting both reputation and revenue. In addition, the social connections give rogue groups a global reach where their attacks spread farther and faster, especially with Twitter which is used extensively by many.

Identifying spammers on Twitter is hard. The problem becomes especially difficult due to resources required to analyze the huge dataset such as that observed by Twitter. The scale of the problem is evident from events such as, say Bin Laden's death, which spurred Twitter users to generate about 12.4 million tweets an hour. Attackers also exploit such opportunities by becoming aggressively active during such events. Manual verification of tweets of such a large scale is unrealistic even on an average day and thus calls for automated detection and prevention approaches. Spammers, in addition, use sophisticated tools which have rendered spam signatures useless. For instance, one such tool used by spammers is the *Spinbot* [11], [1] which generates a sentence with a fixed semantic meaning but varied syntactical structures.

Recent work on detecting spammers has explored methods where spam(mer) detection techniques can be broadly classified into categories: (i) user centric, or (ii) domain or URL centric. The user-centric approaches have analyzed the properties of Twitter users such as the follower-following ratio or distance between the victim and the spammer in the social graph. Such techniques while have been initially useful, have been bypassed by spammers due to their ease of evasion and the limited number of detection metrics. For instance, spammers develop their own network to circumvent the followers-following ratio criterion [12], [11]. Domain or URL centric methods have focused on detecting malicious URLs through honeypot analysis. However, recent malware has been known to disable itself in honeypot environments.

Our technique utilizes a supervised learning based approach where we develop features which distinguish spammers from legitimate users. Our approach banks on converting the strategies adopted by spammers to reach out to large number of

978-1-4673-5494-3/13/$31.00

victims (automation, size of spam) and structural organization of spammers into a lever for detecting spammers. Ours is a hybrid-approach which considers both user and domain/URL centric properties. To evade our methodology, the spammers will have to model human-behavior (which is difficult with current state-of-art) and resort to approaches which will limit their reach, thus making our approach robust. In addition, we do not rely on aggregating extensive organizational information about users or domains/URLs, thereby reducing resource requirement and increasing scalability.

Our main contributions in this work are:

- We develop a set of 15 new features and combine them with three previously proposed features for detecting Twitter based spammers.
- Recognize features that contribute most to spam detection and show that with only 5 tweets, we detect more than 90% of the spammers.
- Through evaluation, we achieve high detection rates with low false positive rates when identifying spammers.
- We cluster the malicious account behavior into spam campaigns to understand the current practices adopted by spammers.

We evaluate our approach on two Twitter datasets containing tweets from more than 600K users. The analysis reveals a stable performance with different supervised learning algorithms where we achieve an average detection rate of 96% with only 0.8% as the false positive rate. We also compare our approach with the state of the art in spammer detection revealing a significant improvement while also highlighting the limitations of the approach used by Twitter itself.

The work in this paper is organized as follows. Section II introduces current techniques for spam detection and summarizes the main features where supervised learning algorithms have been used. We describe our spammer detection system in section III where we also introduce our novel features. The approach is evaluated in section IV while we discuss major sources and tools exploited by spammers to evade existing techniques in V. The limitations and further discussion is highlighted in section VI. We finally conclude and present future directions in section VII.

II. Related Work

A. Spam and Anomaly Detection

To identify distinguishable spammer characteristics, Ramchandran et. al. [33] study the network properties of email spam. Their analysis reveals a correlation between spammers and their physical locality (geographical IP or ASN) while the study also highlights BGP hijacking used for spam attacks. [32] is an extension of ideas from [33] where the authors employ supervised learning using network-level features to distinguish spam from ham. As the work in [29] suggests, Twitter based spam differs qualitatively from email spam, the former being the focus of this paper.

Botnets have used Twitter to post encoded instructions as tweets and embedded URLs for stealing confidential information from victims [2]. Recently, in [3], researchers study how interactions (followings or mentions) between human users varies after introducing social-connector bots which associate two users based on similarity between the posted tweets. User account properties are also studied in [30] where a user's decoy friends are analyzed and the quantitative and qualitative correlation metrics are used to establish if an account has been affected by an online social network worm (such as Koobface or Mikeyy).

B. OSN analysis

1) Statistical Techniques for Spammer Detection: Gao *et. al.* [28] present a technique to detect and characterize spam campaigns on Facebook where connected component analysis of a graph of wall posts with common URLs or similar content is used to identify campaigns. Sarita *et. al.* in [19] study structural properties of legitimate users and spammers and observe similarity between Web graph and Twitter's social graph. Zi Chu *et. al.* [26], analyze behavior of humans, bots and cyborgs on Twitter. According to their observations, bots post more URLs per tweet, mostly use 3rd party APIs for posting tweets, post regularly throughout the day or week while humans tweet less on weekends and nights. In [14] , Koutrika *et. al.* have worked on empirical and comprehensive study of magnitude and implications of spam in tags and how existing tagging systems are vulnerable to spam. Sangho *et. al.* in [27] present an analysis of techniques used by Twitter spammers to avoid detection of URLs by public blacklists and suggest correlating URL redirection chain and user-centric features to detect suspicious URLs in tweets.

Recently, Yang *et. al.* [11] introduced graph based features like local clustering coefficient [9], closeness of a user's neighborhood to a clique, betweenness centrality for spammer detection. Song *et. al.* [21] exploit the fact the spammers are usually not found in close proximity to legitimate users. The proximity is defined as the number of nodes between two accounts in the social graph. These graph based features are difficult to evade but are also time and resource intensive. An interesting study is also carried out in [36] where graphical structures of spam accounts and their emphouter supportive network are studied. The study concludes that the spam community is more tightly knit than legitimate users. Grier *et. al.* [16] note that using public blacklists such as Google SafeBrowsing [4] to examine Domain Names (DNs) and URLs posted would not be useful because of usage of URL shorteners on Twitter and delay in blacklisting a URL.

2) Supervised Learning Approaches: Benevenuto *et. al.* in [13], discuss the rise of video spammers and promoters in video social networks like YouTube. Various features, including video-based, user-based, and social-network based features are used with supervised learning techniques for spammer detection. A large portion of the previous work [18], [16], [22], [24], [23] on Twitter spam detection uses supervised learning approach to build a model of non-benign users based on ground truth and classify users as spam or benign. Note that previous work has relied on user's profile based features like number of URLs posted, followers-to-following ratio, number of mentions, etc. for classification which has not sufficiently addressed the spam problem and can be easily evaded. Our work addresses a larger problem by proposing features which capture the evolution of spammers, rather than focus on detecting the current spamming trend.

Monarch [29] analyzes the Twitter dataset for identifying spam URLs. The real-time detection approach uses L1-regularized Logistic Regression with features relying upon web browser based events, along with DNS, and network properties. We differ by proposing features which capture the *behavior* of spammers, in addition to the infra-structure oriented characteristics of malicious entities, as proposed above. The work in [35] highlights six features, also motivated by the spammer's behavior. The features include sender degree, rate of message sending, or the URLs sent. The features, however, differ in context where one of which (Unique URL number) contradicts our observation.

Our motivation derives from the fact that we can use other Twitter based features to build a faster URL blacklist specifically for Twitter. Different from previous research, our work focuses on devising novel features based on entropy, community-nature of spammers along with our URL blacklist system to attack spammer detection problem on Twitter. We also present an alert system to filter spam tweets in real-time. In addition, we present a study of spam campaigns carried out on Twitter and tactics adopted by spammers.

III. The CATS System

A. Datasets and Ground truth

To evaluate our technique, we use two datasets collected from Twitter at different periods of time. Our first dataset (referred to as dataset A) consists of approximately 500K users with over 14M tweets and about 6M URLs as used in [11]. In [11], the authors collect the dataset by extracting 40 recent tweets for users present in the follower and following set of those accounts whose tweets are observed on the Twitter time line between April 2010 and July 2010. The data contains 2060 spam accounts, identified through an automated as well as manual analysis of posted URLs.

In addition to the above dataset, we also collect a relatively new corpus of 110,789 Twitter accounts (dataset B) collected between November 2011 and January 2012. This dataset B contains 2.27M tweets and 263K URLs. We use three techniques to collect B. We collect tweets and information for 4854 Twitter users, by doing a breadth-first search of the *followings* of verified accounts. The verified accounts are chosen randomly and the breadth-first search spans only one level of the relationship tree.

The second technique used for adding user information to B uses the tweets posted by Twitter accounts. The constant stream of tweets is accessible using the Twitter Streaming API [5] which gives a sample of tweets posted in real-time. For all tweets obtained using the streaming API, we first identify the accounts corresponding to every tweet and then retrieve various statistics of each account (recent 40 tweets posted, followers, following, and other account related features). Lastly, we collect information for the spam accounts using a technique introduced in [21] which involves collecting information of all accounts manually reported as spammers (to @spam) by Twitter users. We checked the status of all reported users and those suspended are added to our ground truth as spam/malicious users. Such a collection adds 407 spam/malicious accounts to our ground truth for dataset B.

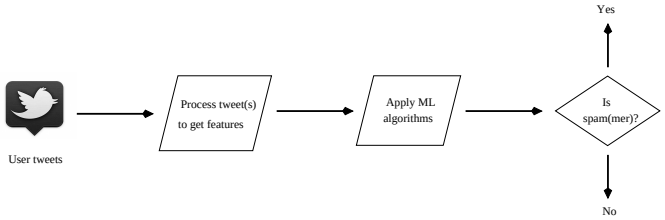


Fig. 1: Components for Twitter based spammer detection.

Note that accounts not identified as malicious are considered benign.

B. Overview

Figure 1 presents the process of classifying a set of users as belonging to a spammer (or otherwise), consequently identifying spammers. The procedure for spammer detection begins with analyzing user's information to calculate feature values that can be fed into a supervised learning algorithm.

Our system, CATS, approaches spam detection from two vantage points. First, it tries to reduce opportunities for the automation of spreading spam. Second, it tries to reduce the chances of a community of spammers appearing to be a normal user community. Both these approaches are based on the observation that, in order to reach a large number of users, the spammers are leveraging automatic/algorithmic approaches and leveraging size (a group of Twitter accounts) and organization (community of accounts with followers and following relationships) to spread spam. However, the need to push the same spam message or URL to several users links these spammers and enables our detection methodology. Our approach turns the size and automation of the spamming campaign into a lever for detecting spammers and to evade our detection methodology, the spammers have to resort to approaches that reduce the reach of their spamming campaigns, which makes our approach robust.

To this end, we propose five categories of features for detection. The first category highlights the *bait-oriented* features which identify how spammers lure victims either by posting spurious tweets with orthogonal content or by intentionally *mentioning* victims in random tweets, hoping that doing so will make the victims read or click on accompanying URLs. The second set of features analyzes how the *behavioral* aspects of spammers differ from benign users. Specifically, we quantify features which help understand the automated approach adopted by spammers to reach a larger scale. We then identify how embedded URLs and the corresponding domains, are distributed in various tweets for a given account, thus identifying a campaign. Another interesting class of features we propose analyzes the content posted by the spammers, and the respective automation in the content of such tweets or URLs. Here we look at the similarity between multiple tweets for an account, or the similarity between tweets and the posted URLs. Finally we also propose looking at the organization of an account's profile where we expect that a well-organized profile is more likely to be benign than malicious.

C. Feature Description

We now describe the motivation behind features used to detect spamming activity. To the best of our knowledge, most features proposed here are novel. For the features that have been previously used, we highlight the source and motivation of using them.

1) Bait-oriented features: This set of features identify the techniques used by spammers to grab a victim's attention or lure the victim into clicking malicious links.

Number of Unique Mentions (NuMn): *NuMn* quantitatively identifies the mention of a Twitter user by an account under analysis. For instance, a tweet "Hello @fred" involves the mention of an account identifier "fred". Benign users repeatedly interact with their relations while spammers mention users randomly [8]. Specifically, we note that a benign account's behavior involves carrying the conversation with a select few accounts which we capture with the *unique* mentions. This heuristic is difficult to evade since it attacks the basic mode of spam distribution on Twitter.

To compute this feature, we simply calculate the total number of unique users mentioned as a fraction of total tweets. More formally:

$$NuMn = \frac{\text{Num of unique users mentioned}}{\text{Total num of tweets}} \tag{1}$$

A high value of this metric (approaching 1.0) indicates that the account is involved in *excessive* mentioning of users and thus its malicious reputation score goes up.

Unsolicited Mentions (UlMn): This feature represents an interesting approach used to compromise victims. We explain using an example. Suppose there is an innocent user *X* followed by a malicious user *Y*. There is no prohibition in *Y* following *X* in the Twitter model. Thus, the spammer *Y*, may mention *X* (using the *@X* tag) in one of his tweets which is seen by *X*. Frequent mentioning of unknown users thus represents malicious intentions.

As in [21], we use the fraction of mentions to non-followers to materialize our observations into a feature. Formally,

$$UlMn = \frac{\text{Num of mentions to non-followers}}{\text{Total num of mentions}} \tag{2}$$

This feature differs from the previously described *NuMn* in that *UlMn* accounts only for the interaction between unrelated users. Since spammers mention users randomly in their tweets who they don't follow, the metric value is expected to be tending to 1.0 for them. Spammers would have to create strong structural connections in order to evade this feature which is difficult.

Hijacking Trends (HjTd):

An interesting phenomena that we observe during our analysis is the way spammers are hijacking trends in an attempt to reach a wider set of users by posting tweets unrelated to the topic (depicted by hash tags). The popular (or trending) topics interest a large number of users which may even visit the corresponding tweets to get more information. Tweeting on the trending topics thus provides a spammer an impetus so that its tweet reaches many accounts and is picked up by an unsuspecting Twitter user. Note that the tweet's content need not necessarily reflect the corresponding topic (hash tag). Thus, we use this feature to determine the similarity between a user's tweets, and the famous tweets observed for the trend. This metric is computed using the cosine similarity measure as defined below:

$$Similarity = cos\theta = \frac{A.B}{\|A\|\|B\|} \tag{3}$$

Here A, B represent two multi-dimensional vectors where each dimension is a unique word that makes up the respective tweet. In equation (3), A represents a tweet that an account under scrutiny makes (containing the trending tag) and B represents one of the 10 famous tweets observed for the trend. We compute the similarity for all tweets and the 10 top trending tweets and average them to obtain *HjTd*. This feature ranges from 0.0 to 1.0. A lower value of this metric implies tweet's textual content is different from the topic of discussion which signifies an attempt to hijack the trend. Spammers could evade this feature by tweeting text relevant to the topic with accompanying URLs instead pointing to spam websites, but this tactic would be caught using features introduced later (as *SiUt*).

Intersection with Famous Trends (FmTd): This attribute extends the *HjTd* feature with an alternate perspective. Using the *FmTd* feature, we evaluate the intersection between the popular trends and the trends that a given user tweets. This is expressed by the equation:

$$FmTd = \frac{\text{Num of trends intersecting with popular trends}}{\text{Total num of trends specified by a user in tweets}} \tag{4}$$

As the value of this metric approaches one, the account is associated with a greater anomaly. The motivation behind using popular trends follows from *HjTd* feature in that those tweets containing popular trends are visible to a larger number of users, increasing the possibility of deceiving users. Note that a reliable indication of spam characteristics can be expected to be visible especially after a prolonged analysis.

2) Behavioral-entropy Features: We now describe features which distinguish the spammers from the benign users, by identifying patterns in their respective activities.

Variance in Tweet Intervals (VaTi): This feature represents the variance (or standard deviation) in the time taken by an account to post tweets. Most spam (bots) have been found to use Twitter APIs or the web interface [26] to post tweets automatically at given intervals. Such a behavior implies lower entropy of the inter-tweet time interval. On the contrary, a legitimate user is expected to tweet stochastically. Therefore, a simple measure incorporating this heuristic is:

$$VaTi = \frac{(X - \mu)^2}{N} \tag{5}$$

where X is the random variable representing the inter-tweet time interval and μ is the mean tweet interval observed for a particular account. N is the total number of tweets minus one, that is, the total values observed for variable X. We associate a low value of this metric with a more

malicious account. Note that this feature value is not upper-bound. We do understand that spammers may bypass this feature by introducing randomization in tweeting times, in which case, other features proposed in this work are expected to assist in detection. The research community may also use the automated tweet observation for introducing preliminary checks for anomaly analysis.

Variance in number of tweets per unit time (VaTw): The *VaTw* feature computes the entropy in the number of tweets that an account posts. We observe that a few spammer post a fixed number of tweets whereas a legitimate accounts post different number of tweets (at different intervals). To compute this metric, we first divide the time line of the user's tweets into bins of different sizes (1 hour, 30 minutes, 20 minutes). We then express the number of tweets posted per bin as a random variable and calculate the variance analogous to equation (5).

Ratio of VaTi and VaTw (TiTw): We use the ratio of features *VaTi* and *VaTw* as another feature for computation. From our dataset, we observe that certain malicious bot accounts tweet in *bursts*. For instance, a bot account may post several tweets within a given unit of time (say one hour) and then sleep for a long time before repeating this activity. We intend to capture this pattern using the ratio of the two previously described metrics. A high feature value indicates that random burst of patterns (with a high variance in the tweet *intervals* while a low variance in *number* of tweets) belong to malicious accounts.

Tweet Sources (TwSc): Tweets can be posted by users through several modes. For instance, users may use the HTTP interface, the API interface, or post tweets from blogging websites. We measure the *different sources* that a particular account may use. A benign user may not necessarily confine posting comments or tweets from a particular source. Spam bots, however, may restrict themselves to select sources due to factors governing scale and automation.

Thus, our metric (*TwSc*) computes the fraction of different sources used for posting tweets. From our ground truth, we observe 200 different sources in total that users utilize for tweeting. A higher value for this feature signifies benign account. As we consider only a maximum of 40 tweets for each user, *TwSc* has the range [0, 1]. Use of multiple tweeting sources is accepted as benign activity and thus higher the value of this metric, the more benign is the corresponding Twitter account.

3) URL Features: The following set of features rely on the posted URL for extracting related information using either the complete URL or only the domain name, as explained below:

Duplicate URLs (DuUr): The duplicate URL feature identifies the number of URLs that are repeatedly tweeted by an account. A spammer posts the same URL over and over again to lure victims into clicking at least one of those malicious links. A legitimate user, however, usually posts on variety of topics, each represented by a different URL. Note that combining the number of mentions (*NuMn*) with the *DuUr* feature gives us a fair indication of spammer's *modus operandi* as spam tweets appear to have both a large fraction of mentions and URLs.

We compute this metric by calculating the average number of times a URL has been posted by a user. More formally, we compute:

$$DuUr = \frac{\text{Num of URLs posted}}{\text{Total num of Unique URLs posted by the user}} \tag{6}$$

Note that the URLs posted within tweets by most accounts are shortened URL strings which we resolve (completely) before computing the above metric. We also note that this feature is robust in the sense that in order to evade detection by this measure, the spammer has to incur extra work/cost in terms of creating multiple URLs with same content. If randomization of URLs is applied to evade detection using this metric, features described later help in detection.

Duplicate Domain Names (DuDn): Similar to *DuUr* metric, *DuDn* identifies the fraction of tweets which contain unique domain names, extracted from a URL. News blog accounts represent obvious false positives as such accounts repeatedly post URLs for the same domain. Such mistakes, however, are discarded when considering other features as discussed here. We elaborate on this aspect in section VI. As an equation, we have:

$$DuDn = \frac{\text{Num of unique domain names in tweets}}{\text{Total num of domain names posted}} \tag{7}$$

Therefore, a value of this metric close to 0 suggests that the account intends to promote a specific domain, a behavior characterizing spammers. For scenarios where multiple spam domains are registered on an IP address, our next feature - IP/Domain fluxing - will detect them.

IP/Domain fluxing (IpDn): The IP-to-Domain ratio is simply the ratio of IPs for the domains (or host names) that are part of the URLs posted by an account. Thus, the two sets, those of IP addresses (denoted by I) and the set of domains D can represent fluxing based on the set cardinalities. Specifically, the metric is:

$$IpDn = \frac{\|I\|}{\|D\|} \tag{8}$$

A high value of *IpDn* reflects IP-fluxing while a low value indicates domain fluxing as many domains point to a few IP addresses. Values of this metric out of the range [1,2] indicates malicious nature with the scale of anomaly dictated by the distance from the range. For instance, legitimate entities usually have one or two IP addresses for each domain name that serves content. several domain names mapping to the same IP address hosting the web content. The possibility, however, of a benign user repeatedly posting CDN domains is also low. Note that domain fluxing is particularly indicative of the malfeasance as has been observed earlier [20].

4) Content-entropy Features: Tweet content of spammers and benign users would evidently be different. We present a number of features based on tweet text content to catch spammers:-

Tweet's language dissimilarity (KlTw): We compute the similarity of an account's tweet to the most widely used

language on Twitter viz. English. The motivation for using this feature comes from the recent observation of botnet activities prevalent on Twitter [2]. We intend to identify such malicious accounts by computing the Kullback-Leibler (K-L) divergence between the alphanumeric character based probability distributions. We use three character distributions for this metric: the distribution obtained from the set of tweets for an account (*test*), the distribution for the English language (*benign*), and a uniform distribution (*malicious*) [20].

The K-L divergence provides a measure of (dis)similarity between two distributions. Thus, to use this metric, we first compute the divergence between the benign distribution (the English language) and the test distribution (denoted by D_g). Next we compute the divergence of the test distribution from the uniform distribution (representing malicious intention), denoted by D_b. Finally we calculate $KlTw = D_g - D_b$ as a feature for the supervised learning algorithm. Note that this feature is expected to perform better with longer and quantitatively larger number of tweets as observed in [20]. Note that while individual divergence values are positive, $KlTw$ is an unbounded variable.

Similarity between Tweets (SiTw): The similarity metric described here identifies the campaign that a particular malicious account pursues. We again use equation (3) (the cosine metric) as a measure of similarity, with dimensional vectors represented by unique words. A higher cosine measure indicates that the account under analysis is perhaps tweeting with similar textual content repeatedly and thus could be a potential spammer. Note that for a set of N tweets, we average the cosine similarity computed between $(N \times (N-1))/2$ unique pairs. It varies between 0.0 and 1.0.

URL and Tweet similarity (SiUt): We further validate the tweets posted by users by checking the content of the tweet and the content of the URL corresponding to the tweet. A spammer might post tweets having text related to interesting events and rogue URLs. For instance, the tweet content referring to a major sports event could land the victim to a web page with pharmaceutical advertisements. Such a feature requires fetching the web page and applying the cosine similarity measure on the tweet's content and the web page content. We finally average the similarity values observed for all tweets containing URLs. A higher value (tending towards a value of 1.0) for this metric, therefore, refers to a more benign account.

5) Profile Features: ***Followers-to-Following Ratio (FrFg):*** The followers-to-following ratio for an account is the ratio of number of followers which are also following, and the number of accounts that the given account is following. It is also represented by:

$$FrFg = \frac{Followers \cap Following}{Following} \tag{9}$$

FrFg is a common and effective metric used by Twitter as well as researchers for spam(mer) identification [24], [19], [18]. Naive spammers attempt to follow many accounts in the hope that the relationship will be reciprocated. The *FrFg* metric addresses this problem by requiring that each account maintain a healthy ratio of their followers and following to avoid being suspended. Note that in order to evade this feature, spammer will have to ensure higher follow-back which is difficult to achieve.

Profile Description's Language Dissimilarity (KlPd): Analogous to using the K-L divergence measure for computing the entropy of alphanumeric characters present in the tweets, we find the divergence between the profile description of an account, from the English language. We develop this heuristic based on the observation from our dataset wherein spammers do not provide relevant or organized information compared to legitimate users. The test distribution represents the alphanumeric characters retrieved from publicly available profile information. We consider the higher divergence as a greater indication of spam.

Table I summarizes the features we use for learning spammers' behavior. Column 1 provides rank of each feature in classification, column 2 provides feature names. In column 3, we note the mutual information (MI) measure identifying the contribution of each feature towards spam identification. Column 4 highlights resource requirements for feature computation - computational resources (C), network resources (N), or both. Finally, we highlight which of the proposed features in this work are novel (new) or discussed previously (old).

IV. Evaluation

We present the evaluation of CATS, by analyzing the ground truth described earlier in section III-A, consisting of 2467 spam accounts and 4854 benign accounts (from verified accounts and their *followings*). We compute feature values for each of these accounts and feed them to four different supervised learning algorithms (utilizing the Weka tool) - Decision Tree (DT), Random Forest (RF), Bayes Network (BN) and Decorate(DE). All performance statistics reported here are based on 10-fold cross validation over the dataset.

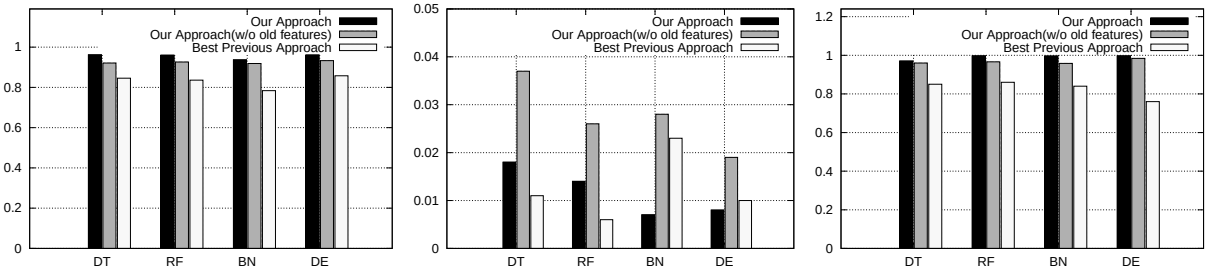
A. Performance Comparison

Figure 2(a) compares the True Positive Rate (TPR) of our approach with an algorithm proposed in [11]. The work in [11] is known to outperform current spam detection techniques on Twitter. To check the effectiveness of our features, we also present performance of our approach using only the novel features that we propose. True Positive Rate (TPR) is defined as the fraction of spammers correctly identified by our algorithm. Similarly, the false positive rate (FPR) denotes classifying a legitimate account as a spammer. From the figure, we observe a consistently better detection rate compared to [11], using different supervised learning algorithms. Specifically, we note a TPR improvement of more than 15% for all classifiers used, with the best performance observed for the Decorate classifier. We also note that our approach can catch 93.6% of spammers with a FPR of 1.8% without using old features.

Figure 2(b) compares False Positive Rate (FPR) of our technique with that of evaluation done in [11]. Compared to the previously proposed technique, we improve the FPR for two of the classifiers, again obtaining the highest scale of improvement with the Bayes Network classifier. That is, we see the FPR reducing from 2.3% to 0.7%. Finally, figure 2(c) summarizes the above results by highlighting that the area under the ROC curve is almost one (the maximum achievable). Note that the ROC curve evaluates the TPR with FPR. Thus,

TABLE I: Features used for classification

Rank	Feature description (code)	MI for ranking	Types of delay	old/new feature
1	Duplicate URLs	0.27	C	new
2	Followers-to-Following ratio	0.26	C	[11]
3	Number of unique mentions	0.21	C	new
4	Unsolicited mentions	0.21	C + N	[21]
5	Duplicate domain names	0.19	C	new
6	Variance in tweet intervals	0.16	C	new
7	Hijacking trends	0.13	C + N	new
8	Tweet's language dissimilarity	0.12	C	new
9	Known Spam URLs	0.12	C	new
10	Ratio of VaTi and VaTw	0.11	C	new
11	IP/Domain fluxing	0.11	C + N	new
12	Known Spam Domain Names	0.11	C	new
13	Variance in number of tweets per time unit	0.11	C	new
14	Tweet sources	0.09	C	new
15	Similarity between tweets	0.08	C	[22], [23], [24], [11], [21]
16	Intersection with famous trends	0.07	C	new
17	URL and tweet similarity	0.07	C + N	new
18	Profile description's language dissimilarity	0.07	C	new



(a) True Positive Rate (b) False Positive Rate (c) ROC Area

Fig. 2: Performance Comparison with previous approach.

greater the area under the curve, the better is the performance. In addition to above evaluation, our detection mechanism using only the novel features achieves an ROC area of 98.4% using the Decorate supervised learning technique.

We note that Twitter has its own policy of identifying and suspending spammers. Figure 3(a) compares latency of detecting spammers using our algorithm with the Twitter suspension algorithm described above. We observe that Twitter identifies malicious activity by spammers later while our approach catches all of them on Day 0. Some spamming accounts are successful in evading suspension for as long as two weeks. We note a possibility of "selection bias" here as we only test a small set of users, whereas Twitter deals with all the users and hence they may have to be more careful about false positives which might result in bad publicity.

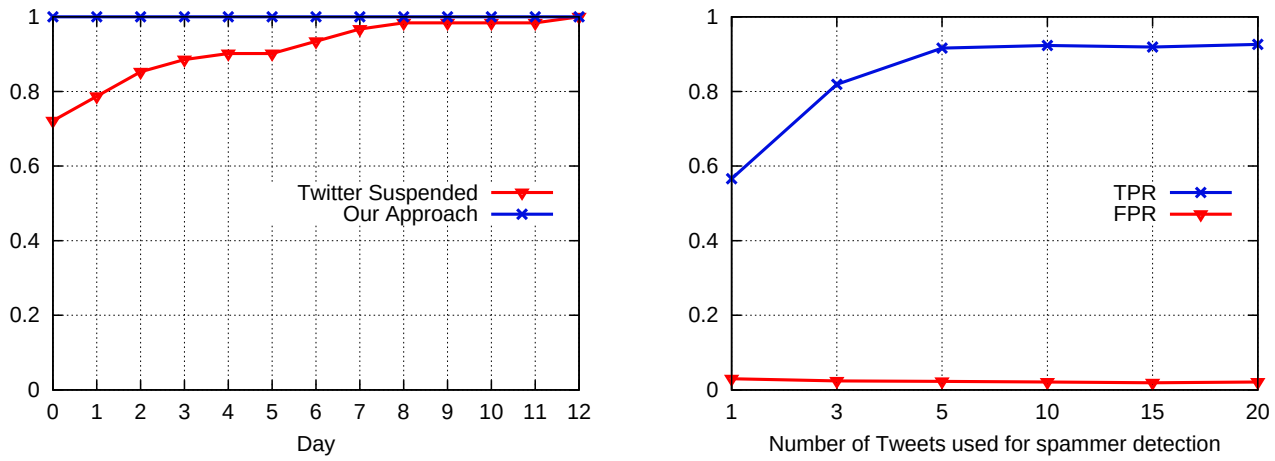
Similar to the experiment above, we analyze the number of tweets required to detect spammers. Figure 3(b) elaborates such an analysis. As the figure shows, our technique detects more than 90% of the spammers with only five tweets and *more than half* of the spammers with only a single tweet posted by each anomalous accounts.

B. Identifying Unknown Spammers

To identify unknown spammers in our dataset, we use a modified ground truth set for training and use the generated model to test a randomly chosen dataset of 31,808 users. Specifically, our ground truth comprises of benign accounts from dataset A and the spam accounts suspended by Twitter (in dataset B). Using the learned model, CATS labels 2378 (7%) accounts as spammers. Compared with CATS, we find that Twitter had suspended only 21.4% of the above 2.3K accounts.

To validate our classification, we select a sample of 238 (10%) users randomly from the above mentioned spam accounts and verify them manually. We also intend to understand the nature of spam campaigns carried out on Twitter.

We broadly categorize the identified spammers into the following classes : spammers suspended by Twitter (TwtSus), spammers promoting/advertising a product (AdSpm), spammers posting links which are either in public blacklists or blacklisted by shorteners (MalLnk), benign users classified as spammers (FlsPos), and users who promote their account to gain followers but haven't posted any spam content (Others). As in Figure 4(a), we note that around 37% of spammers detected by our technique, are also suspended by Twitter, while a major portion of active users are participating in advertisements and posting phishing/malware URLs. Some users, classified as Others, post same tweet to random accounts asking them to follow back to discuss sports events. This could be a new scam to increase followers. Figure 4(a) also shows FPR of our model which is 3.01%. Further analysis reveals that though the accounts posted benign URLs, their tweeting behavior resembled an automated bot.



(a) (b)

Fig. 3: (a) CDF of Spammer Detection given Spammer was eventually suspended by Twitter (Twitter vs Our Approach), (b) Evaluation of spammer classifier varying number of tweets used.

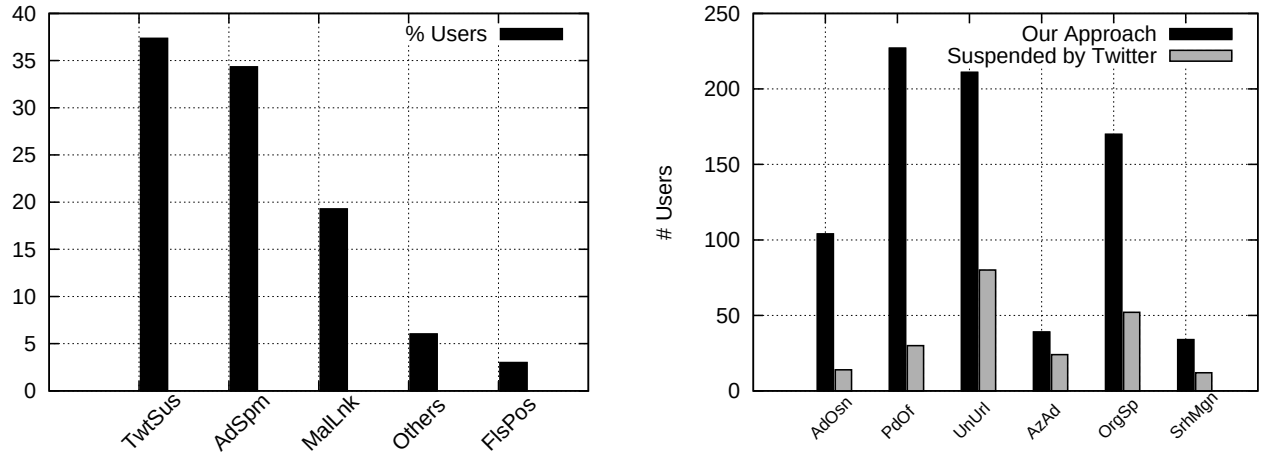


Fig. 4: (a) Analysis of users classified as spam.(b) Detection of users participating in a campaign (Twitter vs Our Approach)

V. Spam Campaigns

To understand and isolate spam behavior among the rogue groups, we do a clustering analysis of the spammers labelled by CATS. The feature vector for clustering utilizes only three attributes, which includes the destination domain names used by a spammer, tweet text content, and the tweet source. We use only the three features based on our manual investigation of a sample of malicious accounts. Based on our investigation, the proposed clustering features clearly differentiate the spamming techniques and therefore used for identifying campaigns as described here. We use k-means clustering algorithm for our analysis. We choose k=15 as the residual error vs k curve is observed to flatten at k=15 which signifies optimal number of clusters for our problem.

A. Campaigns

On clustering the spam accounts with the feature vectors as described earlier, we observe five major groups or campaigns in our dataset with a high density for further analysis. The five campaigns differ primarily in their mode of operation or evasion techniques. In figure 4(b), we highlight the fraction of malicious users detected per identified campaign. We now describe each of the five campaigns.

1) Organized Spammers (OrgSp): We observe a group of spammers which appear to follow a master, emulating a botnet infrastructure where the C2 (Command and Control) server instructs the bots (in our case, Twitter spam accounts) to spread spam. We observe certain bot accounts posting similar messages and the embedded URLs to victims. In addition, such a group of bot accounts is aggressive in following a large number of legitimate users and run the risk of being deactivated due to violation of Twitter terms for the acceptable follower-following ratios. The bot accounts also mention random benign Twitter users in their tweets to entice them, as described earlier in this work.

This campaign first appears in our dataset on December 14, 2011 and some users participating in this campaign are still active. The campaign consists of 170 users, 6672 tweets and 6314 URLs (note that our dataset only has 40 latest tweets). Figure 5(a) shows an instance of spam propagated by

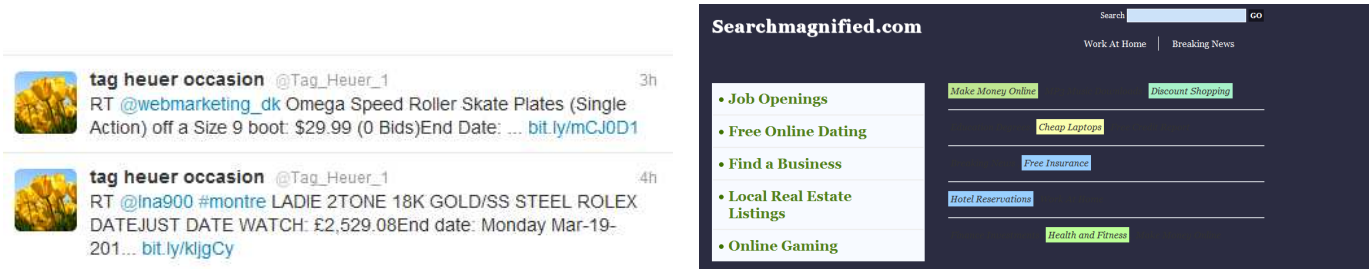


Fig. 5: (a) Users participating in multiple campaigns. (b) SearchMagnified.com Website.

one such group. The organized structure of these community of spammers (master-slave model) makes it easier for the controller to advertise different products.

2) Amazon ad spammers (AzAd): We also observe a large number of users advertising Amazon products on Twitter. Such accounts participate in Amazon's affiliate program which facilitates commission to a user if a product is bought from referrals through tweets or custom URLs. A few accounts post misleading tweets containing news report headings of the day as text and post URLs which point to Amazon's products. Such campaigns raise questions over the grey area of definition of spam in general. Note that a similar campaign is also reported in [12].

Our cluster contains 41 such users who posted a total of 1280 tweets and 1281 URLs. A survival strategy observed among accounts in this cluster is that several of these accounts have a large number of followers, which we hypothesize as being purchased. Such accounts themselves follow a large number of accounts and thus maintain the threshold follower-to-following ratio to avoid being suspended by Twitter.

3) URL-based tactics to evade Twitter (UnUrl): We find that a cluster of users post URLs which when visited respond differently to different HTTP user-agents. For instance, a visit to a spam URL using conventional probe tools (such as *wget*) does not reveal the page, as against the case where the same URL is visited through a web browser.

More specifically, following anomalous patterns have been observed with such URLs:

- Bad http response (503, 403 to API but spam page to browsers).
- Returning different URL based on HTTP request agent.
- URL no longer valid.
- Domain name no longer available.
- Blogger deleted either the post or the posting account, or perhaps was already identified as spreading spam. (tumblr.com is used most)

The spammers anticipate use of non-conventional browsing tools for detection and therefore modify their behavior to increase evasion capability. Such tactics used by spammers are also observed in [27]. Note that the corresponding cluster contains about 348 such users posting a total of 11,458 tweets and 8970 URLs.

4) Spamming using other OSNs (AdOsn): A large volume of spam URLs on Twitter refer to web links posted on other major social media like Facebook. These campaigns highlight the click-jacking behavior where ad clicks earn revenues for the malicious groups. Several web-based content sharing platforms like YouTube, blogger.com are also major contributors of spam in Twitter. Video-based spam which is outlined by promotional videos of newly launched products, advertisements, etc., is a novel way of spamming. It would be challenging for the existing methodology to detect such spammers since most of the techniques factor URL or domain reputation for spam detection.

The cluster for such spammers contains 36 users who posted 802 tweets and 747 URLs. The spammers mention Twitter users randomly to reach out to a larger set of benign users. They mention a total of 846 users (which is more than one mention per tweet) out of which only 435 are unique. We note that this implies weak co-ordination amongst spammers. We have 65 users posting links to advertisement already posted on Facebook containing a total of 1327 tweets and 1261 URLs. These spammers employ various tactics to disseminate spam - some hijack famous trends like #iphone4, #iphone4s, etc., while some mention random users. These users post unrelated tweets using many tags - a total of 824 times on 80 different tags.

5) Spam via searchmagnified.com (SrhMgn): A large portion of spam constitutes tweets with the embedded domain as *searchmagnified.com* which is a drive-by download website. It was detected sometime back when a browser hijacker redirected all the visited websites - to *searchmagnified.com* (Figure 5(b)). These tactics show a clear trend of attackers turning to Twitter for a greater reach. Also, Twitter users are more likely to click-through since the URLs are obfuscated through URL shortening services and is posted by a follower. We find 34 such spam accounts posting a total of 770 tweets and 711 URLs. Most of these users mention random users to disseminate spam to benign Twitter users. On an average, we see 0.88 mentions per tweet posted by these users.

B. Comparison with Twitter

Figure 4(b) shows twitter does well to catch users advertising amazon products, links which are blocked by shorteners, links caught by some online blacklist service like Google SafeBrowsing. But our approach does better in catching new evolving spam content like *searchmagnified.com*, spammers using Facebook and YouTube for spam dissemination. Our approach does better in detecting users acting as advertisers of product offers on various websites (PdOf) like etsy.com. Twitter suspends only a small portion of users whose URL content is missing. Twitter's approach to spam detection needs to get more aggressive as the spammers get more aggressive in reaching out to benign users.

VI. Discussion

While we highlight a good performance using our features when compared to existing techniques for spam detection, we also note that spammers may modify their own evasion techniques. For instance, individual features proposed in this work, may be altered in a fashion which resembles the activity of a legitimate account. However, we attack basic strategies used by spammers to spread spam vigorously both in terms of size and organization. Thus, evading our system would require spammers to resort to strategies which will confine the reach of spam campaigns, thus making our system robust.

We note that previous approaches have proposed methods with a smaller set of features for detecting spammers [21]. Such an approach, and several from the past have explored the social graph and quantified metrics based on who and how is a spammer connected to other users. We note that collection of such an extensive (and sufficiently informative) social graph of an account is non-trivial. On the contrary, our technique utilizes a modest number of tweets for a given account for analysis, making the deployment of our approach scalable.

In this work, we have designated spam emanating tweets as spammers. While we do acknowledge that legitimate accounts on Twitter may be compromised by rogue groups for malicious activities, we believe that such compromised accounts should also be labeled malicious and duly addressed as they pose threat to the larger connected community.

VII. Conclusions and Future Work

In this paper, we propose several features for spammer detection on Twitter. We introduce features which exploit the behavioral-entropy, profile characteristics, bait analysis, and the community property observed for modern spammers. The features are largely dependent on easily retrievable information resulting in minimal latency. We use tactics used by spammers to reach out and organized spamming against them. Our evaluation with a previous best known technique highlights the improvement in both detection rate and the corresponding false positive rate resulting in a good system performance. Additionally, we highlight detecting more than half of the spammers with only a single tweet post. We also identify prevalent spam campaigns using unsupervised learning algorithms, in an attempt to better understand the mode of operation of spammers.

As a future work, we plan to expand upon the category of baits used to compromise victims. Therefore, we plan to further categorize the trends that the spammers use to be able to distinguish them from legitimate users. Additionally, we plan to use evolving campaigns as indicators for the proliferation of spam, and use it as an antidote for anomalies.

References

[1] Spin Bot. http://www.spinbot.com/

[2] Jose Nazario. Twitter Based Botnet C&C. http://ddos.arbornetworks.com/2009/08/twitter-based-botnet-command-channel/, 2011

[3] Twitter Bots Create Surprising New Social Connections. http://www.technologyreview.com/web/39497/page1/ *MIT Technology Review*, Jan. 2011

[4] Google SafeBrowsing API. http://code.google.com/apis/safebrowsing/

[5] Twitter Development API. https://dev.twitter.com/

[6] F-measure. http://en.wikipedia.org/wiki/F1_score.

[7] Search Magnified Website. http://www.searchmagnified.com/

[8] Twitter Mentions. https://support.twitter.com/articles/14023-what-are-replies-and-mentions/

[9] Local Clustering Coefficient. http://en.wikipedia.org/wiki/Clustering_coefficient/

[10] Bin Laden's Death Sparks Record 12.4 Million Tweets Per Hour. http://mashable.com/2011/05/02/bin-laden-death-twitter/

[11] Chao Yang, Robert C. Harkreader, and Guofei Gu. Die Free or Live Hard? Empirical Evaluation and New Design for Fighting Evolving Twitter Spammers. In *Recent Advances in Intrusion Detection (RAID)*, 2011.

[12] Kurt Thomas, Chris Grier, Vern Paxson, and Dawn Song. Suspended Accounts in Retrospect: An Analysis of Twitter Spam. In *ACM Internet Measurement Conference (IMC)*, 2011.

[13] Benevenuto, Fabrício and Rodrigues, Tiago and Almeida, Virgílio and Almeida, Jussara and Gonçalves, Marcos. Detecting Spammers and Content Promoters in Online Video Social Networks. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, 2009.

[14] Koutrika, Georgia and Effendi, Frans Adjie and Gyöngyi, Zoltán and Heymann, Paul and Garcia-Molina, Hector Combating spam in tagging systems. In *Intl Workshop on Adversarial Information Retrieval on the Web (AIRWeb07)*, 2007.

[15] Anirudh Ramachandran, Nick Feamster, and Santosh Vempala. Filtering Spam with Behavioral Blacklisting. In *ACM Conference on Computer and Communications Security*, 2007.

[16] Chris Grier, Kurt Thomas, Vern Paxson, and Michael Zhang. @spam: The Underground on 140 Characters or Less. In *ACM Conference on Computer and Communications Security*, 2010.

[17] Daniel Gayo-Avello and David J. Brenes. Overcoming Spammers in Twitter A Tale of Five Algorithms. In *Spanish Conference on Information Retrieval (CERI)*, 2010.

[18] Benevenuto, Fabricio and Magno, Gabriel and Rodrigues, Tiago and Almeida, Virgilio. Detecting Spammers on Twitter. In *Collaboration, Electronic messaging, Anti-Abuse and Spam Conference*, 2010.

[19] Sarita Yardi, Daniel Romero, Grant Schoenebeck, and Danah Boyd. Detecting spam in a twitter network. *First Monday 15(1)*, 2010.

[20] Sandeep Yadav, Ashwath K.K. Reddy, A.L. Narasimha Reddy, and Supranamaya Ranjan. Detecting Algorithmically Generated Malicious Domain Names. In *ACM Internet Measurement Conference (IMC)*, 2010.

[21] Jonghyuk Song , Sangho Lee, and Jong Kim. Spam Filtering in Twitter using Sender-Receiver Relationship. In *Recent Advances in Intrusion Detection (RAID)*, 2011.

[22] Kyumin Lee, James Caverlee, and Steve Webb. Uncovering Social Spammers: Social Honeypots + Machine Learning. In *ACM SIGIR conference on Research and development in information retrieval*, 2010.

[23] Alex Hai Wang. Dont follow me: Spam Detection on Twitter. In *Proceedings of 5th International Conference on Security and Cryptography (SECRYPT)*, 2010.

[24] Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. Detecting Spammers on Social Networks. In *Annual Computer Security Applications Conference (ACSAC)*, 2010.

[25] Benevenuto, Fabrício and Rodrigues, Tiago and Cha, Meeyoung and Almeida, Virgílio Characterizing user behavior in online social networks. In *ACM Internet Measurement Conference (IMC)*, 2009.

[26] Zi Chu, Steven Gianvecchio, Haining Wang and Sushil Jajodia. Who is Tweeting on Twitter: Human, Bot, or Cyborg? In *Annual Computer Security Applications Conference (ACSAC)*, 2010.

[27] Sangho Lee and Jong Kim. WARNING BIRD: Detecting Suspicious URLs in Twitter Stream. In *Network & Distributed System Security (NDSS)*, 2012.

[28] Gao, Hongyu and Hu, Jun and Wilson, Christo and Li, Zhichun and Chen, Yan and Zhao, Ben Y. Detecting and Characterizing Social Spam Campaigns. In *ACM Internet Measurement Conference (IMC)*, 2010.

[29] Kurt Thomas and Chris Grier and Justin Ma and Vern Paxson and Dawn Song. Design and Evaluation of a Real-Time URL Spam Filtering Service. In *IEEE Security and Privacy*, 2011.

[30] Xu, Wei and Zhang, Fangfang and Zhu, Sencun. Toward worm detection in online social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference*, 2010.

[31] Cristian Lumezanu and Nick Feamster and Hans Klein. #bias: Measuring the Tweeting Behavior of Propagandists. In *Proceedings of the AAAI International Conference on Weblogs and Social Media (ICWSM)*, 2012.

[32] Shuang Hao and Nadeem Ahmed Syed and Nick Feamster and Er G. Gray and Sven Krasser. Detecting Spammers with SNARE: Spatio-temporal Network-level Automatic Reputation Engine. In *Proceedings of the USENIX Security Symposium*, 2009.

[33] Ramachandran, Anirudh and Feamster, Nick Understanding the network-level behavior of spammers In *SIGCOMM Comput. Commun. Rev.*, 2006.

[34] Provos, Niels and Mavrommatis, Panayiotis and Rajab, Mohbeeb Abu and Monrose, Fabian. All your iFRAMEs point to Us. In *ACM Security Symposium*, 2008.

[35] Hongyu Gao, Yan Chen, Kathy Lee, Diana Palsetia and Alok Choudhary. Towards Online Spam Filtering in Social Networks. In *Proc. of 19th Network & Distributed System Security Symposium (NDSS)*, 2012.

[36] Chao Yang, Robert Harkreader, Jialong Zhang, Suengwon Shin, and Guofei 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)*, 2012.