SPAMMER DETECTION ON ONLINE SOCIAL NETWORKS

A Dissertation

by

AMIT ANAND AMLESHWARAM

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

MASTER OF SCIENCE

Approved by:

Co-Chairs of Committee, AL Narasimha Reddy
Riccardo Bettati
Committee Members, Guoefi Gu
Department Head, Hank Walker

December 2012

Major Subject: Computer Science

ABSTRACT

Twitter with its rising popularity as a micro-blogging website has inevitably attracted attention of spammers. Spammers use myriad of techniques to lure victims into clicking malicious URLs. In this thesis, we present several novel features capable of distinguishing spam accounts from legitimate accounts in real-time. The features exploit the behavioral and content entropy, bait-techniques, community-orientation, and profile characteristics of spammers. We then use supervised learning algorithms to generate models using the proposed features and show that our tool, spAmbush, can detect spammers in real-time. Our analysis reveals detection of more than 90% of spammers with less than five tweets and more than half with only a single tweet. Our feature computation has low latency and resource requirement. Our results show a 96% detection rate with only 0.01% false positive rate. We further cluster the unknown spammers to identify and understand the prevalent spam campaigns on Twitter.

DEDICATION

To my parents and sisters

ACKNOWLEDGEMENTS

I would like to sincerely thank Dr. Reddy for giving me the opportunity to work with him. I believe that working here has prepared me to deal with future challenges in a much better way. I would like to thank Dr. Bettati and Dr. Gu for being on my committee.

I would also like to thank my colleagues at the lab, especially Sandeep. Finally, I am indebted to my parents and sisters for their constant support and motivation.

TABLE OF CONTENTS

Page

ABSTRACT ii

DEDICATION iii

ACKNOWLEDGEMENTS iv

TABLE OF CONTENTS v

LIST OF FIGURES vii

LIST OF TABLES viii

1. INTRODUCTION 1
   - 1.1 Basic Introduction 1
   - 1.2 Our Contribution 3

2. RELATED WORK 5

3. OVERVIEW 10
   - 3.1 Twitter - Rise of a Social Media 10
   - 3.2 Spammers 11
     - 3.2.1 Nature of Spam 12
     - 3.2.2 Tools of the Trade 12
   - 3.3 Problem Definition 14

4. SPAMBUSH - OUR SYSTEM 15
   - 4.1 Machine Learning Approach 15
   - 4.2 Datasets and Ground Truth 15
   - 4.3 Overview 17
   - 4.4 Feature Description 18
     - 4.4.1 Bait-oriented Features 18
     - 4.4.2 Behavioral-entropy Features 21
     - 4.4.3 URL Features 23
     - 4.4.4 Content-entropy Features 24
     - 4.4.5 Profile Features 26
     - 4.4.6 Community-based Features 27

5. EVALUATION 31
5.1 Performance Comparison 31
5.2 Identifying Unknown Spammers 33
5.3 Real-time Tweet Classification 35

6. ANALYSIS OF SPAMMER CAMPAIGNS 38
6.1 Campaign of Spammers 38
6.1.1 Organized Spammers 38
6.1.2 Amazon Ad Spammers 39
6.1.3 URL-based Tactics to Evade Twitter 39
6.1.4 Spamming using other OSNs 40
6.1.5 Spam via searchmagnified.com (SrhMgn) 41
6.2 Comparison with Twitter 42

7. DISCUSSION 43

8. CONCLUSION & FUTURE WORK 44

REFERENCES 45

LIST OF FIGURES

FIGURE Page

4.1 Components for Twitter based spammer detection. 17

4.2 Components to update Spam URLs occurrences. 27

5.1 Performance Comparison with previous approach. 32

5.2 Comparison of Spammer Detection Approaches (Twitter vs ours). . . 33

5.3 Analysis of users classified as spam. 34

5.4 Tweet classification Results (a) with 60% tweets unclassified, (b) varying % of unclassified tweets from 5% to 90%. 36

6.1 User participating in multiple campaigns. 38

6.2 SearchMagnified Website 41

6.3 Users participating in Campaign(Twitter vs Our Approach) 42

LIST OF TABLES

TABLE Page

4.1 Features used for classification 30

1. INTRODUCTION

1.1 Basic Introduction

Online Social Network (OSN) are websites where users can create profiles, establish connection with other users and converse with them. There are hundreds of such OSN websites present today. Facebook, Twitter, etc. are the most popular ones boasting more than 500 million active users. Twitter, as an Online Social Network, is intended to help people converse using text-based posts called tweets. Popularity of Twitter and other OSNs has been rising in recent times having played crucial role in connecting people and providing a discussion forum on several occasions like protests in Syria[9].

Spammers are users on Twitter which analogous to e-mail spam try to spread malicious content or advertise using their social network on Twitter. With rise of Twitter as Online Social Network, it has inevitably attracted large number of spammers. OSNs have been fighting spammers since their inception. In August 2009, Twitter observed around 11% of tweets posted were spam. Twitter has its social structure built by users following each other posts which in turn signifies trust between users. This along with millions of users provide a perfect platform for spammers to disseminate spam.

Spammers not only try to advertise products, they have also been actively involved in deceiving users into clicking malicious links. Spammers on Twitter employ myriad of techniques to lure users into clicking malicious URLs. Techniques which deceive users into clicking such URLs include but are not limited to: befriending (to *follow* in Twitter terminology) unrelated users and sending unsolicited messages. To gain a wider reach to potential victims, spammers are known to befriend (to *follow* in Twitter terminology) unrelated users, send unsolicited messages and masquerade malicious components (for instance, using URL shorteners to substitute *malicious-appearing* URLs.), to convince the victim of their legitimacy.

Preventing spam proliferation translates to protecting users from clicking malicious links. The malicious URLs pose threats in the form of drive-by-downloads and other infections. 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. From Twitter's perspective, spam threatens to prohibit the growth of user base hurting both reputation and revenue.

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. An example of such a scale comes from the event where Bin Laden's death spurred Twitter users to generate about 12.4 million tweets an hour. Spammers, in addition, use sophisticated tools which have rendered spam signatures useless. One such tool used by spammers is the *Spinbot* [14, 1] which generates a sentence with a fixed semantic meaning but varied syntactical structures.

There has been various approaches introduced both in industry and academia to fight spammers on Twitter. Engineers at Twitter Inc. has been working actively to control spammers. They have introduced a set of rules which dictates the behavior of users on Twitter[10]. They define behavior which would be considered as spammer and such accounts are suspended by Twitter. According to Twitter rules, users following/un-following aggressively, having lower follower-following ratio, posting duplicate content are considered spammers. However Twitter mentions that definition of spamming behavior will keep on changing depending upon new tricks used by spammers.

Twitter also has a user where spammers could be reported, however those users are further analyzed by Twitter policies before being suspended. 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 for 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 seen adaptation by spammers. For instance, spammers develop their own network to circumvent the followers-following ratio criterion [15, 14]. 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.

1.2 Our Contribution

Our technique utilizes a supervised learning based approach where we develop features which distinguish spammers from legitimate users in *real-time*. More specifically, we detect 56% of the total spammers with 3% false positive rate using single posted tweet. Additionally, we show that we detect *all* spammers within the first day compared to Twitter which takes more than two weeks. Our feature set for detecting spammers essentially exploits the entropy characteristics which distinguish spammers from others. Our approach banks on converting the strategies adopted by spammers to reach out to large number of 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. In order 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 3 already used features for detecting Twitter based spammers.
- Recognize features that contribute most to real-time spam detection.
- 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 a Twitter dataset 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. In addition, the clustering analysis reveals how spammers form organized groups and utilize a community of spammers to assist in spam proliferation or evading detection.

The work in this thesis is organized as follows. Section 2 introduces current techniques for spam detection and summarizes the main features where supervised learning algorithms have been used. We present a formal definition of Twitter and its features in Section 3. Here, we will also formalize problem of spammer detection on Twitter. We describe our spammer detection system in Section 4 where we also introduce our novel features. The approach is evaluated in Section 5 while we discuss major sources and tools exploited by spammers to evade existing techniques in Section 6. The limitations and further discussion is highlighted in Section 7. We finally conclude and present future directions in Section 8.

2. RELATED WORK

Spammer detection on Online Social Networks is difficult not only because of grey nature of spam in general but also due to spammers trying to adept to existing techniques. Various OSNs like Facebook, Youtube etc. has been targeted by spammers to reach out to users. OSNs provide a perfect platform for spammers to disguise as a benign user and try to get malicious posts clicked by normal users. Benevenuto *et. al.* in [16] discuss rise of video spammers and promoters in video social networks like YouTube. They analyze users's behavior on Youtube and propose some features which could distinguish spammers from normal users and use supervised learning techniques to detect spammers and promoters on Youtube. Various features, including video-based, user-based, and social-network based features are presented. Video-based features like number of views, comments, number of ratings etc. capture properties of the uploaded video. User-based features like number of friends, videos watched, videos uploaded etc. give an idea about the uploader. Various social-network based features like clustering coefficient etc. try to distinguish spammers from benign users based on social relationship between friends. Gao *et. al.* [28] present a technique to detect and characterize spam campaigns on Facebook. They collect an anonymized dataset of wall posts from Facebook and analyze them to identify spam campaigns on Facebook. They try to form a graph using the wall posts - adding an edge between posts with similar content or containing same destination URLs. Connected components in the graph signify similar posts contents by different users. Then they use bursty nature and geographically distributed nature of spammers in order to identify connected components which participate in spam campaigns.

Recently, researchers in [2] have identified Twitter accounts participating as Botnet Command & Control (C&C) server. The involved account posted base64 encoded intructions for bots as tweets. They post URLs from which bots can download updated DDLs which act as information stealers on victim's system. Recently, in [3], researchers study how interactions (following, mentions) between other human users change after introduction of social-connector bots which introduce two users based on similarity between the posted tweets. These bots monitor all the tweets posted by users on Twitter and mention two users who are tweeting about similar topics in past. This is the bot's way of introducing two Twitter users having similar interests.

Sarita *et. al.* in [20] study structural properties of legitimate users and spammers and observe similarity between Web graph and Twitter's social graph. They hypothesize that normal users are at the center of social graph (following each other and some celebrities), celebrities are at one end (mostly being followed by normal users) while spammers lie at the other end following a lot of normal users. Zi Chu *et. al.* [26] analyze behavior of humans, bots and cyborgs on Twitter. According to their observations, bots post more URLs per tweet, post regularly throughout the day or week while humans tweet less on weekends and nights. They also observe that bots mostly post tweets using API-based tools while humans mostly use web interface for tweeting. They also note that bots have larger number of followings as compared to followers, while humans have similar number of followers and followings. Cyborgs, on the other hand, have larger followers than followings. In [17], 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. They model tagging systems, spam in tagging systems. They describe methods to rank documents in the tagging system and present effects of inclusion of passive moderators on the tagging system. Sangho *et. al.* in [27] present an analysis of techniques used by Twitter spammers to avoid detection of URLs by public blacklists. They suggest various URL based features like length of redirection chain, features based on correlating URL redirection chain like position of entry point URL in redirection chain, number of different initial URLs etc. They also use various featurs based on tweeting user - standard deviation of account creation date, number of different Twitter accounts posting the URL, text similarity for tweets posting the URL. They consolidate spamming behavior of users on Twitter along with URL based features to classify a URL spam or legitimate.

With rise of spam in Twitter, many approaches have been proposed for spam detection in Twitter. A large portion of previous work [19, 23, 25, 24] 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. Earlier work focussed on finding words posted in tweets which appeared in popular spam blacklists and various content based features like number of URLs per word in tweet, number of hashtags per word, number of mentions per word etc. [19] *et. al.* also used user behavior based features like age of the user, number of followers, number of times the user was mentioned, number of times the user was replied to, number of times the user replied someone etc. In [18], authors study nature of URLs posted by spammers and observe that only a small part of posted phishing/malicious links are identified by public blacklists. They cluster the users posting blacklisted URLs in order to study how users participate in spam campaigns. Lee *et. al.* in [23] use similar user-centric features along with tweet text similarity for latest 20 tweets, bi-directional friend's ratio to identify spammers on Twitter. Previous work has mostly 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. A large portion of previous work on spam detection on Twitter used followers-to-following ratio. To evade this feature, spammers have used various tactics including buying followers from various services, following each other to maintain a healthy ratio.

Recently, Yang *et. al.* [14] introduced graph based features like local clustering coefficient [8], closeness of a user's neighborhood to a clique, betweenness centrality for spammer detection. They also use API-based features like number of URLs posted using 3rd party API etc along with graph-based features. Their main motivation for graph-based features is its difficult to evade them since spammers will have to form large number of friendship relationship with benign users. Song *et. al.* [22] 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. Grier *et. al.* [18] 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.

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.

In order to generate ground truth, most of previous works use honeypot approach[23]. They introduce various legitimate accounts and an associated bot on Twitter as honeypot. These accounts gather information (profile information, tweets, followers) about users who try to follow them. Lee *et. al.* find that this strategy can attract spammers employed on Twitter for various purposes. Other group of researchers use blacklisting services like Capture-HPC, Google Safe Browsing API[14] to analyze URLs posted and label users posting a larger portion of blacklisted URLs as spammers. Few researchers have also tried users suspended by Twitter[15] or users reported to Twitter's official account @spam[22] to gather their ground truth.

3. OVERVIEW

This section will describe the Twitter domain and formally define various aspects of Online Social Networks like Twitter and how they are exploited by spammers. We will also define the problem formally.

3.1 Twitter - Rise of a Social Media

Twitter is the fastest growing Online Social Network (OSN) today. Twitter, as other OSNs, is intended to let people converse to each other about their topics of interest. It has approximately 500 million users registered[11]. Twitter is unique in its structure and functionality. It is a micro-blogging website. Users registered can send text-based posts of upto 140 characters called Tweets. A total of around 340 million tweets are generated everyday [11]. Basically, a tweet can contain text and links to URLs explaining the ideas of the tweet. Users have to be concise in their posting because of limit of 140 characters, thus forming it as informal chit-chat with friends. It would be impossible for a user to track all tweets posted on Twitter, so Twitter has various features which helps narrowing down the tweets of interest to a user. Every user has a timeline which contains tweets which might be interesting to the user.

Twitter has concept of *following.* By definition, if a user A follows user B, it signifies that all tweets posted by B would be posted on timeline of user A. The concept of *following* is directed meaning if A follows B, B will not see the tweets posted by A. Now, A could also follow B, making the friendship relation between A and B. This feature helps in specifying users whose tweets a user would be interested in. These users could be a friends, co-workers, celebrities or famous researchers.

Since Twitter is an OSN, it has to have social aspects as well. Recently, it has been acting as a news social-media spreading breaking news around the globe. Twitter has *trending topics* which can be seen on left of the timeline when a user logs in. Trending topics contain top 10 hot topics being discussed in the area selected. Trends are specific to location selected, which could be changed by the user. Twitter identifies topics which are hot currently rather than on daily basis, which help users discover recent activity on their location of interest. Trending topics could be a phrase, name of celebrity etc. In order to post a tweet in a *trending topic*, user has to include # before the term while posting a tweet. For example, if you want tweet some text in trend JustinBeiber, you will have to post #JustinBeiber in the tweet content. A hashtag is a token in tweet followed by # symbol. Now, hashtags called also be used by users interested in some topic to form a virtual group on Twitter. Note that users interested in the topic would have to post same hashtag in their posts.

A user might communicate with other by posting a tweet directly to him/her. Twitter provides *mentions* which does exactly that. In order to mention a user, you would have to post followed by screen name of the user whom the post is intended for. The user mentioned will see the tweet even though it is not following the user posting the tweet.

We need to understand these techniques of Twitter because these are exploited by spammers to spread spam as is explained later.

3.2 Spammers

Anyone who is familiar with internet has faced spam of some sort, be it e-mail spam, spam on forums, newsgroups etc. Spam is defined as use of electronic messaging system to send unsolicited bulk messages. With the rise of OSN, it has become a platform for spreading spam. Spammers on Twitter are users who try to send unsolicited messages to a large audience with the intention of advertising some product or infecting user's system.

3.2.1 Nature of Spam

In this section, we will discuss nature of spam disseminated on OSN. As in e-mail spam or any other mode of spam, spammers have only one intention in OSN : making money. Spammers intend to post advertisements of products to unrelated users. As is discussed in [15], some spammers post legitimate links to news posts which are shortened by shorteners which display advertisement before re-directing to destination URL. These spammers make money proportional to number of users clicking the shortened URL. Some spammers post URLs which are phishing websites which steal user's sensitive data. Spammers also post URLs which download malware into user's system and make it part of a bot-herd. A botnet can take part in spreading e-mail spam, carrying out distributed Denial-of-service (DDoS) attacks or even stealing sensitive information from user's systems. In recent times, we have seen some Twitter users participating as Botnet C&C servers posting new commands and instructions in their timeline[2].

3.2.2 Tools of the Trade

As discussed in previous section, the main motive of spammers is to make money. In order to make more money, spammers will have to deceive large number of users into clicking the posted links. In order to reach out to large number of victims, spammers on OSN, specifically Twitter, employ a myriad of tools which we are going to discuss in this section. First of all, spammers try to reach out to thousands of benign users, so that they may deceive a large number of users into clicking the links. In order to achieve this, spammers use a variety of Twitter functionalities discussed previously in this section.

One such tool is to *follow* a large number of users randomly. The intention is that a portion of benign users will follow back. These following users will see all posts by the spammer. Recently, various researchers have used follower-following ratio in order to catch such spammers. But spammers have adopted various techniques, like buying followers, following each other etc. to maintain a higher follower-following ratio, to evade such features.

Other famous tool for spam-dissemination famous among spammers is sending direct messages to victims. As discussed previously, spammers can *mention* random Twitter users and the mentioned users would see the post in his/her timeline. This is analogous to sending unsolicited e-mails. It is easier to deceive a user using *mention* because of trust placed by users in OSN. Note that the spammer need not make relationship with victims to spread spam this way.

Another tool used by spammers to disseminate spam is using *trending topics*. Spammers post malicious links into currently trending topics using *hashtag*. Twitter makes it really easy to get current trends by exposing them via API[5]. We need to note here that these posts are exposed to a large number of users who would be interested in the topic. Also, there need not be any relationship between victim and spammers, which makes little room for suspicion and make benign users vulnerable to these posts.

These are basic spam dissemination strategies used by spammers currently. However, we note that with the introduction of various other features by Twitter, spammers will use evolved tactics for spreading spam. Also, since OSNs are relatively new, spammers on OSN might be less sophisticated, but they are evolving their skills to evade the current state-of-art in spammer detection.

3.3 Problem Definition

So, designing a system to fight spam on Twitter boils down to protecting users from clicking malicious links. This could be done by stopping users from posting tweets which contain such malicious links at run-time or detecting and removing users repeatedly making such posts.

In this thesis, we are going to attack both the problems. We will present a system which can be used to detect spammers offline. We will also present a *real-time* system which will filter spam tweets as they are posted.

4. SPAMBUSH - OUR SYSTEM

In this section, we will discuss our system - spAmbush which can be used to detect spammers on Twitter offline. We will start from how we collected data from Twitter, describe how we labelled a subset of data collected into ground truth. Then, we will discuss how we use the ground truth to build a model for spammers on Twitter.

4.1 Machine Learning Approach

We are going to use *Machine Learning* approach to detect spammers. Machine Learning requires various datasets - trainset, testset. The trainset is used to build a model for spammer/benign users. An account on Twitter is represented by a feature vector which signifies various characteristics of the account based on which the user is classified as spammer/benign.

4.2 Datasets and Ground Truth

Now, we will describe how we collected our dataset(s). We use two datasets collected 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. In [14], 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 timeline between April 2010 and July 2010. This data was analyzed for malicious/phishing URLs using public blacklists like Google SafeBrowsing [4] and PhishTank[12]. The URLs were also checked by Capture-HPC - a high interaction honeypot client to identify websites which host malware. Users posting more than 10% spam tweets are classified as spammer. Such users are manually analyzed and a set of 2060 spammers is generated.

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 about 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. Note that malicious accounts could follow celebrities (verified account owners) to feign account legitimacy and thus improve their chances of spreading spam. For instance, a victim may identify common interests (in terms of who they follow) with a spam account thereby being convinced into accessing spam URLs. Since we collect only users being followed by verified users, this subset is likely to be free from spammers.

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 mentioned API, we retrieve information of accounts (recent 40 tweets posted, followers, following, and other account related features) which post the corresponding tweets. This approach comprise the largest share of the dataset B.

Lastly, we collect information for the spam accounts using a technique introduced in [22]. The approach involves collecting information of all accounts reported as spammers (to @spam). The reports (complaints) are registered by users to Twitter by mentioning them in tweets posted to @spam, an official twitter profile for reporting suspicious 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.

Our ground truth comprises of 4854 benign users collected by doing breadth-first search of *followings* of verified users. Spammers in the ground truth comprise of 2060 spammers analyzed from DataSet A along with 407 users reported to Twitter and suspended by them. Hence, a total of 2467 spammers and 4854 benign users constitute our ground truth.

4.3 Overview

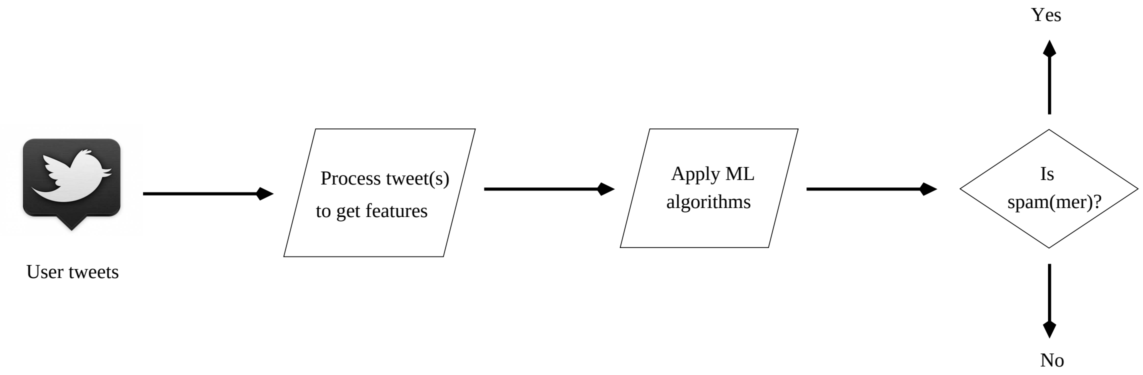


Figure 4.1: Components for Twitter based spammer detection.

Figure 4.1 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, spAmbush, 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.

Previous studies have pointed to the difficulty of using blacklists to detect spam on Twitter (because of long lag times in populating blacklists) and this motivated our work on using Twitter based features to build a faster URL blacklist and use it in *real-time* classification of tweets. We present a method to classify tweets based on several features and show that our technique can be very effective in initial screening of tweets. We use our observations that in order to reach out to large number of victims and to lure them into clicking links, spam tweets behave differently from benign tweets. Our approach exploits these strategies used for spamming along with the URL blacklist to catch spam tweets.

4.4 Feature Description

4.4.1 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): Usually, benign users repeatedly converse with their relations and spammers mention [7] victims randomly. Note that a benign account's behavior involves carrying the conversation with a select few accounts which we capture with the *unique* mentions. This feature is difficult to evade since it attacks the basic mode of spam distribution on Twitter.

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

$$NuMn = \frac{\text{\# of Unique Users mentioned}}{\text{Total \# of tweets}} \tag{4.1}$$

A high value of this metric 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. Using an example, suppose there is an innocent user *Neo* and a malicious user *AgentSmith*. The spammer *AgentSmith*, however, may mention *Neo* (using the *@Neo* tag) in one of his tweets which is seen by *Neo* and perhaps lures him to click on the URL posted in the tweet. Frequent mentioning of unknown users thus represents malicious intentions.

As in [22], we use the fraction of mentions to non-followers to materialize our observations into a feature. Formally,

$$UlMn = \frac{\text{\# of mentions to non-followers}}{\text{Total \# of mentions}} \tag{4.2}$$

Since spammers mention users randomly in their tweets who they don't follow, the metric would be higher 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 audience. The trend on Twitter represents the most popular topics that users are tweeting/discussing about. Each user can attach a hashtagged word which summarizes the tweet.

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 (hashtag). We observe that spammers try to reach out to larger audience by posting tweets unrelated to the topic. 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:

$$Similarity = cos\theta = \frac{A.B}{\|A\|\|B\|} \tag{4.3}$$

Here A, B represent two multi-dimensional vectors where each dimension is a unique word that makes up the respective tweet. In equation (4.3), 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*. 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 but posted URLs 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:

$$FmTd = \frac{\text{\# of trends intersecting with popular trends}}{\text{Total \# trends specified by a user in tweets}} \tag{4.4}$$

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.

4.4.2 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): *VaTi* is another feature we develop considering the automated behavior of spammers. The 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:

$$\text{VaTi} = \frac{(X - \mu)^2}{N} \tag{4.5}$$

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.

Variance in number of tweets per unit time (VaTw): The *VaTw* feature computes the entropy in the number of tweets that an account posts. The idea derives from the *VaTi* feature which looks at the variance in the time taken to post tweets.

With regard to the *VaTw* feature, a spammer's account will post a fixed number of tweets whereas a legitimate account is expected to post different number of tweets (at different intervals). To compute this metric, we first divide the timeline 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 (4.5).

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 pattern. 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.

4.4.3 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:

$$\text{DuUr} = \frac{\text{\# of URLs posted}}{\text{Total \# of Unique URLs posted by the user}} \tag{4.6}$$

Note that we use destination URL to compute this 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 is employed to make them unique, later described features will 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 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 7. As an equation, we have:

$$DuDn = \frac{\text{\# of unique domain names in tweets}}{\text{Total \# of Domain Names posted}} \tag{4.7}$$

Therefore, a value of this metric close to 0 suggests that the account intends to promote a specific domain, a behavior characterizing spammers. We note that this feature can be evaded by registering multiple DNs to same IP address, but this tactic would be caught by our next feature - IP/Domain fluxing.

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:

$$IpDn = \frac{\|I\|}{\|D\|} \tag{4.8}$$

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. Note that domain fluxing is particularly indicative of the malfeasance as has been observed earlier [21].

4.4.4 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*) [21].

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.

Similarity between Tweets (SiTw): The similarity metric described here identifies the campaign that a particular malicious account pursues. We again use equation (4.3) (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.

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 is this metric, therefore, refers to a more benign account.

4.4.5 Profile Features

We now present some features which use profile information of users to separate spammers from benign users.

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:

$$FrFg = \frac{Followers \cap Following}{Following} \tag{4.9}$$

FrFg is a common and effective metric used by Twitter as well as researchers for spam(mer) identification [25, 20, 19]. 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 that we use 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.

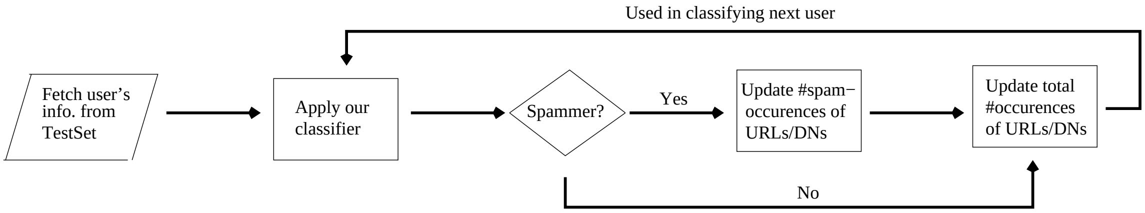

Used in classifying next user
Fetch user's info. from TestSet
Apply our classifier
Spammer?
Yes
Update #spam−occurences of URLs/DNs
Update total #occurences of URLs/DNs
No

Figure 4.2: Components to update Spam URLs occurrences.

4.4.6 Community-based Features

As we discussed earlier, recently spammers started forming community in order to reach out to large number of spammers. Each member of community posts to random users, thus they might co-ordinate to reach-out to large number of victims. We use features which converts such attempts by spammers into a tool to detect them.

Known Spam URLs (KsUr): Let PsUr denote the prior probability of a URL to be spam calculated from a collection of URLs in ground truth. Equation (4.10) calculates PsUr by finding fraction of times a URL is posted by spammers. Higher value of PsUr suggests that the URL appears frequently in tweets posted by spammers thus implying its spam nature. Since a URL keeps changing with the campaign, we keep the list of URLs and their corresponding PsUr value updated with newly caught spammers. Figure 4.2 explains the flow of how the list of spam URLs and their frequencies are updated with each new spammer detected.

$$\mathsf{PsUr(u)} = \frac{\#\text{ of occurrences of } u \text{ in spam database}}{\text{Total }\#\text{ of occurrences of URLs}} \tag{4.10}$$

To compute the metric KsUr, we find average of PsUr values of all URLs posted by a user.

$$\text{KsUr} = \frac{\sum\limits_{\text{all URLs posted}} \text{PsUr(u)}}{\text{Total \# of URLs posted}} \tag{4.11}$$

The motivation behind this feature comes from the fact that organized spammers tend to spread same spam URLs to larger audience using thousands of spam accounts.

Known Spam Domain Names (KsDn): Similar to KsUr, we create a list of known Domain Names (DNs) and their probability of being spam (PsDn), as given by equation (4.12), from ground truth. We keep updating the list of DNs and their corresponding PsDn with newly caught spammers as shown in figure 4.2. Equation (4.12) calculates probability of a domain name, *DN* being spam (denoted by *PsDn*). by finding fraction of times *DN* is used by spammers.

$$\text{PsDn}(dn) = \frac{\text{\# of occurrences of } DN \text{ in spam database}}{\text{Total \# of occurrences of } DN} \tag{4.12}$$

To compute *KsDn*, we find average of PsDn values of all DNs posted by a user. If a user posts DNs each having a high probability of being spam, the user is classified as a spammer.

$$\text{KsDn} = \frac{\sum\limits_{\text{all DNs posted}} \text{PsDn(dn)}}{\text{Total number of DN posted}} \tag{4.13}$$

These two features would evolve themselves with time and cope with changing spam campaigns in Twitter as we keep the frequency information of URLs/DNs updated. As is discussed in [15], a spam campaign can be kept alive by creating new users for months even after a few rogue accounts are caught. The above discussed two features will help in identifying the whole community and burying the campaign once only a few of the spammers are caught.

Table 4.1 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. The computed value also indicates the scale of each feature's contribution, relative to other features. Column 4 highlights resource requirements for feature computaion - computational resources (C), network resources (N), or both. Note that a feature relying heavily on network resources, affects the latency in classifying spam due to relatively greater delay in retrieving information required to compute the attribute value. As we note in table 4.1, only 4 of 18 features we propose, use network resources. Finally, we highlight which of the proposed features in this work are novel (new) or discussed previously (old).

Table 4.1: Features used for classification

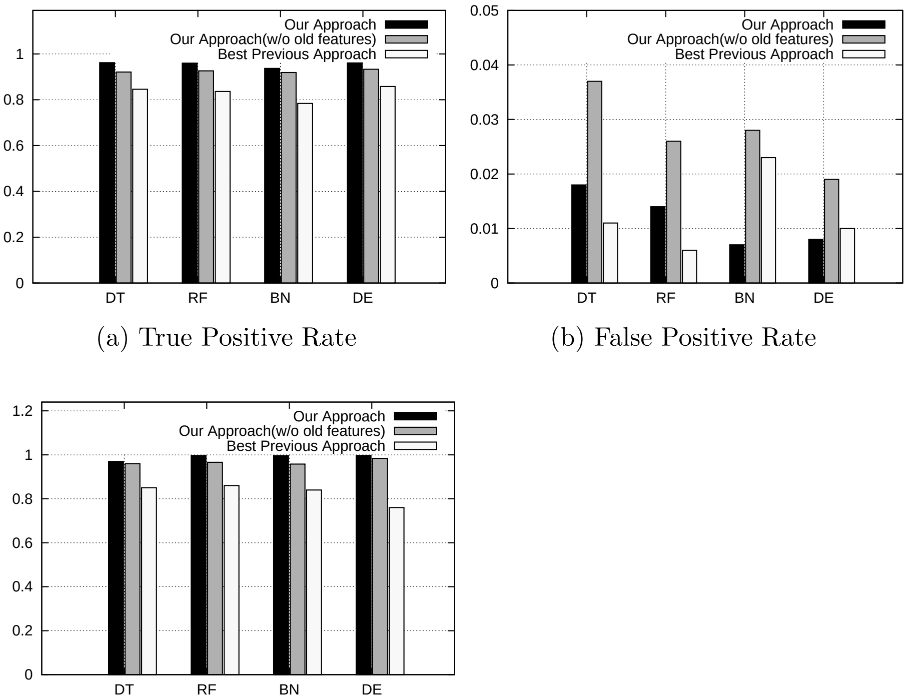
Rank	Feature description	MI for ranking	Types of delay	old/new feature
1	Duplicate URLs	0.27	C	new
2	Followers-to-Following ratio	0.26	C	[14]
3	Number of unique mentions	0.21	C	new
4	Unsolicited mentions	0.21	C + N	[22]
5	Duplicate domain names	0.19	C	new
6	Variance in tweet intervals	0.16	C	new
7	Hijacking 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	[23, 24, 25] [14, 22]
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

5. EVALUATION

We present the evaluation of spAmbush, by analyzing the ground truth described earlier in section 4.2, 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 - Decision Tree (DT), Random Forest (RF), Bayes Network (BN) and Decorate(DE). We used Weka[13] toolkit, a collection of machine learning algorithms, for evaluation of our approach. We present results achieved by these four supervised learning algorithms since they achieve the best classification results on the ground truth. Also, we chose these supervised learning algorithms to compare our approach with existing solutions. All performance statistics reported here are based on 10-fold cross validation over the ground truth (or trainset).

5.1 Performance Comparison

Figure 5.1(a) compares the True Positive Rate (TPR) of our approach with an algorithm proposed in [14]. The work in [14] 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 [14], 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



(a) True Positive Rate

(b) False Positive Rate

(c) ROC Area

Figure 5.1: Performance Comparison with previous approach.

features.

Figure 5.1(b) compares False Positive Rate (FPR) of our technique with that of evaluation done in [14]. 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 5.1(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, 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 previously noted that Twitter users can flag certain accounts as spammers which are then validated by Twitter's approach. If found guilty of spreading spam, the corresponding accounts may then be suspended. Figure 5.2(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 5.2(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.

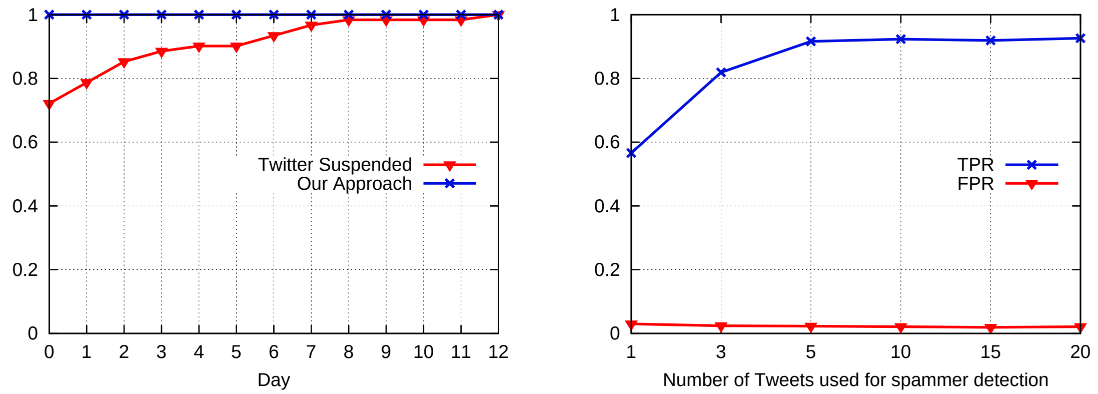


Figure 5.2: Comparison of Spammer Detection Approaches (Twitter vs ours).

5.2 Identifying Unknown Spammers

To identify unknown spammers in our dataset, we use a modified ground truth set for training and use the generated model. Specifically, our ground truth comprises of benign accounts from dataset A and the spam accounts suspended by Twitter (in dataset B) and identified in dataset A. Since some of our features are dependent on language of the tweets, we filter out users which post only English tweets for our analysis. We pick such 31,808 users from our dataset for further analysis. In order to classify them as spammer/benign, we calculate all the features for a user and feed them into our system. We used Decorate for this experiment because it delivers best cross-validation results. Using the learned model, spAmbush labels 2378 (7%) accounts as spammers. Compared with spAmbush, 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.

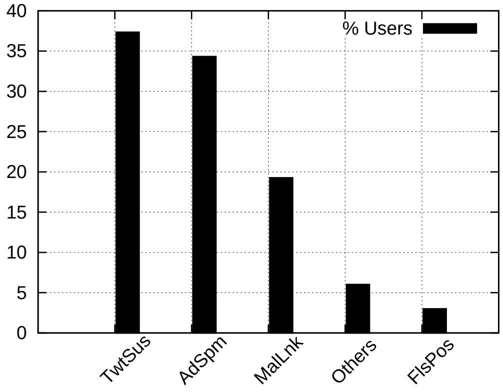


Figure 5.3: Analysis of users classified as spam.

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 5.3, 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 5.3 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.

5.3 Real-time Tweet Classification

Twitter is already heavily burdened with its daily increasing usage. Analysis of each single user needs tweets and user information which overburdens Twitter database. In order to avoid this, we built a system which can classify all incoming tweets into three classes - spam, legitimate and unknown. This can be used not only in alerting users about possible spam in tweets as it is posted, it can also be helpful in filtering users which could be sent to spAmbush model for further analysis.

We build a training set by handpicking 60 users suspended by Twitter which haven't posted any legitimate tweets. Tweets posted by such users are classified spam. Then we pick 220 high-profile verified users from DataSet A and tweets posted by them are classified legitimate. A total of 1902 spam tweets and 8169 legitimate tweets comprise trainset. Similarly, we build a testset containing 417 spam tweets and 1660 legitimate tweets.

We attack a novel problem of classifying tweets as spam/legitimate at real-time using a set of 11 features broadly categorized as community-based, url-based, text-based and bait-oriented :-

- Known Spam URLs.

- Known Spam DNs.
- Similarity between tweet text and URL content.
- Total number of mentions in tweet.
- Source of tweet.
- Total number of hashtags in tweet.
- Tweet's text language dissimilarity.
- Text similarity between popular tweets of a trend and posted tweet's text.
- Total number of famous trends posted in tweet.
- Total number of URLs posted in tweet.
- Fraction of posted URLs present in public blacklists.

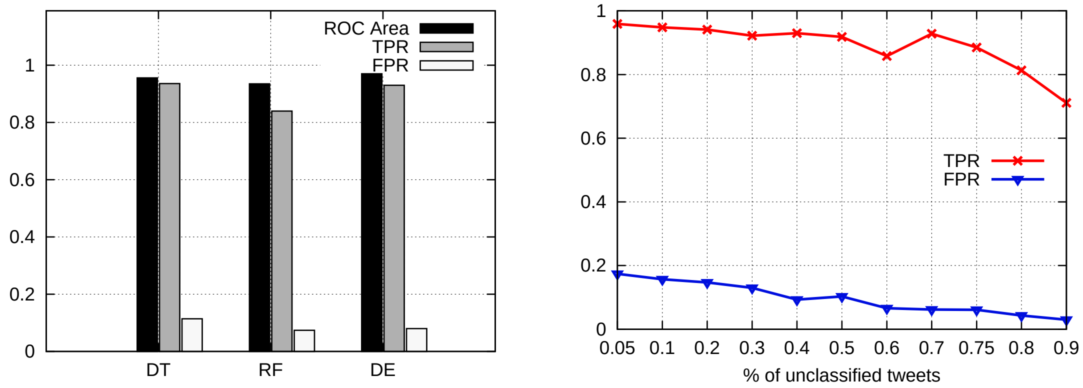


Figure 5.4: Tweet classification Results (a) with 60% tweets unclassified, (b) varying % of unclassified tweets from 5% to 90%.

These features are presented in decreasing order of their importance in classification of tweets. As we can see, community-based features are most important in tweet classification. This could imply presence of users participating in spam campaigns in our dataset.

Since we have restricted amount of information in a tweet, some of the tweets have little evidence of being either spam or legitimate. Due to presence of these tweets which can't be classified as spam or legitimate with confidence, the performance of our approach suffers. To resolve this problem, we classify outlier tweets in our trainset as "unknown". The tweets which can't be classified in either of existing classes are classified as unknown. We assume that each feature for spam and benign users follow normal distribution. Note that a feature's value for spam and benign users are drawn from different distributions - both of which are normal but different mean and variance. Since the feature random variables are independent, the feature vector follow multi-variate normal distribution. We find Mahalanobis Distance (MD) for each tweet's feature vector from the mean of features' distribution and classify tweets lying far from mean as "unknown".

$$MD(x) = \sqrt{(x-\mu)^t \Sigma^{-1} (x-\mu)} \tag{5.1}$$

Figure 5.4(a) presents evaluation of our tweet classification approach with 60% tweets classified as "unknown". We can classify 93% of tweets correctly as spam with a FPR of 6.6% using Decorate supervised learning algorithm.

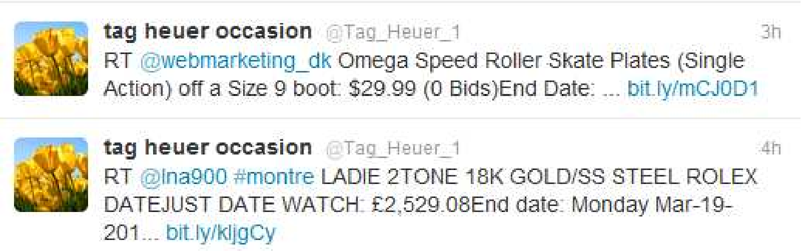
In order to study the effect of percentage of unclassified tweets (tweets classified as unknown) on performance of our approach, we vary the % of unknown-class tweets from 5% to 90% and present performance evaluation of our approach in Figure 5.4(b). We can achieve FPR as low as 3% but with a compromise of percentage of spam tweets detected (lowered to 71.1%). We note that best results are seen with 60% unclassified tweets - catching 93% of spam with FPR of 6.6%.

6. ANALYSIS OF SPAMMER CAMPAIGNS

We clustered and analyzed the users detected by our approach. Clustering users based on what they are spamming and mode of spam was a difficult task because of presence of randomness and varying tactics amongst users. Various techniques used by spammers to hide themselves in huge online social networks which make it difficult to cluster them. We intend to study the state-of-art and outreach of spammers campaigns on Twitter. We cluster spammers by destination domain names posted, tweet text content, and tweet source. The motivation for these three features is that a campaign is primarily defined by the domains and tweet content, while tweet sources may help us identify the mode of operation of a set of bots. We use k-means clustering for our analysis. We choose k=15 for our clustering analysis because the residual error vs k curve flattens at k=15 which signifies optimal number of clusters of a set of documents. Note that we vary k from 1 to 50. We present a few campaigns executed on Twitter by spammers identified using spAmbush.

6.1 Campaign of Spammers

6.1.1 Organized Spammers



tag heuer occasion @Tag_Heuer_1 3h
RT @webmarketing_dk Omega Speed Roller Skate Plates (Single Action) off a Size 9 boot: $29.99 (0 Bids)End Date: ... bit.ly/mCJ0D1

tag heuer occasion @Tag_Heuer_1 4h
RT @lna900 #montre LADIE 2TONE 18K GOLD/SS STEEL ROLEX DATEJUST DATE WATCH: £2,529.08End date: Monday Mar-19-201... bit.ly/kljgCy

Figure 6.1: User participating in multiple campaigns.

Some campaigns are organized and appear to be controlled by a centralized party. A cluster of such users constantly follow the controller and re-tweet all the tweets posted by it. This depicts the sophisticated and organized nature of community of spammers in Twitter. Such accounts reach out to legitimate users by following them aggressively hoping they would follow back. Some malicious accounts from this cluster were observed to use the traditional approach of mentioning random users to disseminate spam. This provides a perfect platform for spam as a service.

This campaign first appears in our dataset on Dec. 14, 2012 and some users participating in this campaign are still active. The campaign consists of 170 users, 6672 tweets and 6314 URLs (Our dataset has only 40 latest tweets). The organized structure of these community of users made it easier for the controller to advertise different products as is shown in Figure 6.1.

6.1.2 Amazon Ad Spammers

A large number of users participate in advertising Amazon products on Twitter. Large volume of users participate in Amazon's affiliate program which provides commission to a user if a product is bought from referrals through tweets or custom URLs. Some of these accounts post tweets containing news report headings of the day as text and post URLs which point to Amazon's products. Such campaigns raise a question over grey area of definition of spam in general. Similar spam cluster was reported in [15].

Our cluster contains 41 such users who posted a total of 1280 tweets and 1281 URLs. A survival strategy present among accounts in this cluster is that these accounts have a large number of followers, which we hypothesize as being purchased.

6.1.3 URL-based Tactics to Evade Twitter

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 removed.
- Domain name no longer available.
- Blogger deleted either the post or the posting account. (tumblr.com is used most)

Recently, [27] reported some of these tactics used by spammers to avoid being detected by Twitter. Note that the corresponding cluster contains about 348 such users posting a total of 11,458 tweets and 8970 URLs. Such techniques used by spammers make analysis of posted URLs difficult.

6.1.4 Spamming using other OSNs

A large volume of spam URLs on Twitter refer to web links posted on other major social media like Facebook. Other web-based content sharing platforms like YouTube, blogger.com are also major contributors of spam in Twitter. Video-based spam which constitutes of promotional videos of newly launched products, advertisements etc. is a novel way of spamming. It would be challenging for the existing techniques to detect such spammers since most of the existing 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 coordination 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 try various tactics to disseminate spam - some hijack famous trends like #iphone4, #iphone4s, etc., some try mentioning random users. These users post unrelated tweets using many tags - a total of 824 times on 80 different tags.

6.1.5 Spam via searchmagnified.com (SrhMgn)

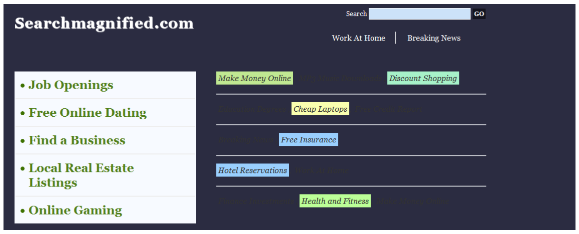


Figure 6.2: SearchMagnified Website

A large portion of spam constitutes of *searchmagnified.com* [6] which is a spam website trying to lure users into visiting the URL which results in installation of spyware/adware. It was detected sometime back when a browser hijacker redirected all the visited websites - to *searchmagnified.com* (Figure 6.2). These tactics show a clear trend how online community trying to compromise systems are now turning to Twitter for a greater reach. Also, Twitter users are more likely to click-through since the URLs are shortened and it is posted by someone in their following list. 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.

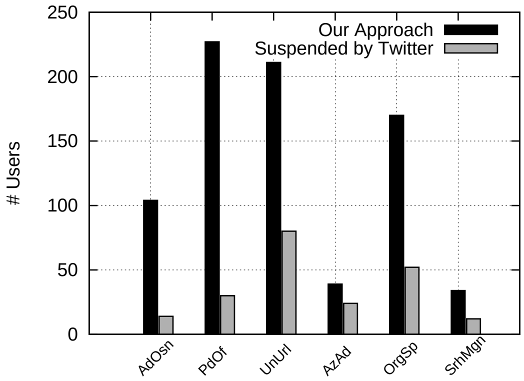


Figure 6.3: Users participating in Campaign(Twitter vs Our Approach)

6.2 Comparison with Twitter

Figure 6.3 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.

7. 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 methodologies to evade detection. 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 [22]. Such an approach, and several from the past have explored the social graph and quantified metrics based who and how is a spammer connected to other users. We note that collection of an extensive (and sufficiently informational) 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.

8. CONCLUSION & FUTURE WORK

In this thesis, 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/, Aug. 2011

[2] Jose Nazario. Twitter Based Botnet C&C. http://ddos.arbornetworks.com/2009/08/twitter-based-botnet-command-channel/, Aug. 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/, Jan. 2012

[5] Twitter Development API. https://dev.twitter.com/, Jan. 2012

[6] Search Magnified Website. http://www.searchmagnified.com/, Feb. 2012

[7] Twitter Mentions. https://support.twitter.com/articles/14023-what-are-replies-and-mentions/, Sept. 2011

[8] Local Clustering Coefficient. http://en.wikipedia.org/wiki/Clustering_coefficient/, Aug. 2011

[9] Syria's spontaneously organised protests. http://www.bbc.co.uk/news/world-middle-east-13168276, March 2012

[10] The Twitter Rules. https://support.twitter.com/articles/18311-the-twitter-rules, Oct. 2011

[11] Twitter On Track For 500 Million Total Users By March. http://www.mediabistro.com/alltwitter/twitter-active-total-users_b17655, March 2012

[12] PhishTank. http://www.phishtank.com/, March 2012

[13] Weka Toolkit. http://www.cs.waikato.ac.nz/ml/weka/index.html, March 2012

[14] Chao Yang, Robert C. Harkreader, and Guofei Gu. Die Free or Live Hard? Empirical Evaluation and New Design for Fighting Evolving Twitter Spammers. In *RAID*, pp. 318-337, 2011

[15] Kurt Thomas, Chris Grier, Vern Paxson, and Dawn Song. Suspended Accounts in Retrospect: An Analysis of Twitter Spam. In *IMC*, pp. 243-258, Nov. 2011

[16] F. Benevenuto, T. Rodrigues, V. Almeida, J. Almeida, and M. Gonalves. Detecting Spammers and Content Promoters in Online Video Social Networks. In *SIGIR*, pp. 620-627, 2009.

[17] 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)*, pp. 57-64, 2007.

[18] 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*, pp. 27-37, 2010.

[19] Fabrcio Benevenuto, Gabriel Magno, Tiago Rodrigues, and Virg lio Almeida. Detecting Spammers on Twitter. In *Collaboration, Electronic messaging, Anti-Abuse and Spam Conference*, 2010.

[20] Yardi, S., Romero, D. Detecting spam in a twitter network. *First Monday 15(1)*, pp. 7-14, 2010.

[21] Sandeep Yadav, Ashwath K.K. Reddy, A.L. Narasimha Reddy, and Supranamaya Ranjan. Detecting Algorithmically Generated Malicious Domain Names. In *IMC*, pp. 48-61, 2010

[22] Jonghyuk Song , Sangho Lee, and Jong Kim. Spam Filtering in Twitter using Sender-Receiver Relationship. In *RAID*, pp. 301-317, 2011

[23] Kyumin Lee, James Caverlee, and Steve Webb. Uncovering Social Spammers: Social Honeypots + Machine Learning. In *SIGIR*, pp. 435- 442, 2010.

[24] Alex Hai Wang. Don't follow me: Spam Detection on Twitter. In *Proceedings of 5th International Conference on Security and Cryptography (SECRYPT)*, pp. 1-10, July 2010.

[25] Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. Detecting Spammers on Social Networks. In *ACSAC*, pp. 1-9, 2010.

[26] Zi Chu, Steven Gianvecchio, Haining Wang and Sushil Jajodia. Who is Tweeting on Twitter: Human, Bot, or Cyborg? In *ACSAC*, pp. 21-30, 2010.

[27] Sangho Lee and Jong Kim. WARNING BIRD: Detecting Suspicious URLs in Twitter Stream. In *NDSS*, pp. 62-74, 2012.

[28] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Zhao. Detecting and Characterizing Social Spam Campaigns. In *IMC*, pp. 35-47, 2010.