SPAM: A Framework for Social Profile Abuse Monitoring

Ayon Chakraborty, Jyotirmoy Sundi, Som Satapathy
(alphabetically ordered)

Department of Computer Science, Stony Brook University
{ aychakrabort, jsundi, srsatapathy, } @cs.sunysb.edu

Abstract—Online social networks have evolved over the last few years with new sophisticated features to help make a user’s virtual world more realistic. Unfortunately, with every new social feature a new privacy security threat often opens up. It’s a good rule of thumb to only connect and share with people you know in real life. “Friending” strangers or making your personal data available to strangers by allowing them to follow you opens you up to added privacy and security risks. According to a study from Cloudmark, nearly 40 percent of new Facebook profiles are fake, created by malware writers and spammers. Even a large social network like Twitter is challenged everyday by presence of multitude of fake Twitter users. They are there not to interact, but just to have a free platform to spam the system with their commercials. They collect as many as possible followers in order to spam them.

In this project we have proposed and evaluated a framework for social network privacy protection via automatic abusive user detection. We have also proposed a 4-class classification model for measuring profile similarity indexing based on fine-grained user similarity features. This would provide a measure of confidence based on which a user can filter out a potential threat. Our Abuse detection model is performing with an precision of 90.9% and the novel profile similarity indexing model is performing with a precision of 64.9% for 4-star, 53.8% for 3-star, 53.8% for 2-star and 58.8% for 1-star. These results are bound to improve with a larger dataset.

I INTRODUCTION

Online social networking sites such as Facebook, Twitter, GooglePlus, MySpace and LinkedIn allow millions of users to meet new people, stay in touch with friends, establish professional connections and more. As more and more of these social networks come online they have become an apt playground for spammers all over the world and are becoming a new kind of threat to regular users of these sites. Spammers have utilized Twitter as the new platform to achieve their malicious goals such as sending spam, spreading malware, hosting botnet command and control (C&C) channels, and performing other illicit activities. There can be a fine line of difference how people define spam accounts. In this study we have classified such kind of accounts into three categories:

a) Monotonic - Users posting same content multiple and huge number of times.
b) Fake – Users who impersonate other valid users.
c) Abusive – Users who post abusive contents, including harmful urls, porn urls, phishing links and divert away regular users.

Of the above the last category, viz., abusive users is the major and common threat in online social networks. The main contribution of this work is to monitor and protect authentic users from getting victimized by abusive users.

In this paper, we describe, SPAM, an algorithm for Social Profile Abuse Monitoring – an approach to help regular users quickly identify abusive users and guide them about whom to connect with. Let Bob be an authentic user and another user Alice sends him a connection request. The sole intention of this work is to protect the Bobs of a social network from the influence of abusive users. We take a two step approach for the SPAM algorithm. First it will check Alice’s profile information whether it’s a genuine one or has abusive contents. Then it will match the profile similarity between Bob and Alice and come up with a recommendation for Bob whether or not to accept Alice’s request. The first portion of the algorithm, which is detecting profiles with abusive contents help us to filter out a lot of abusive users in the first place before even comparing the profiles for a match. SPAM takes as input two profiles, the sender of the connection request and the recipient. It outputs to the request recipient four possible kinds of recommendation: four stars, three stars, two stars and one star, meaning respectively, high match, moderate match, low match and almost no profile similarity. It also indicates whether the profile was detected for abusive contents. Our algorithm basically acts as a filter to warn users, based on which users can take a better decision. We employed several machine learning techniques like Support Vector Machine, Naïve Bayes Classifier, Decision Tree and Random Forests. Of them Support Vector Machine performed the best classification and the results were encouraging concerning recent literature. We obtained an accuracy of 89% in abusive profile detection. To the best of our knowledge this is the first work where we have integrated abusive profile detection and profile similarity measurement to get an overall index for the recommendation. In this study we have selected the Twitter social network for evaluation and testing our framework but the results could be applicable to any social networks in general. We used profile data of 5000 Twitter users for our evaluation and experiments. These were classified as authentic and abusive user profiles.

The rest of the paper is organized as follows. Section II describes the related works in this context. Details about the data collection procedure are described in Section III. We discuss the choice of features and modeling in Section IV.
Evaluation, results and Analysis are presented in Section V. In Section VI we make some comments about the performance and accuracy of SPAM algorithm followed by conclusion in Section VII.

II RELATED WORK

Due to the popularity of Twitter, many studies have been conducted with an aim at studying detection of abusive profiles / spam tweets etc. Also independent work has been done in Link Prediction and Profile similarity between users. Some of the important work in this domain is listed below: [1] describes a set of 24 features for fake profile identification. They have also classified the profiles using various machine learning algorithms and compared them with each other. In [2] most of the work has been done to protect different user groups in social networks from being flooded by spammers advertising their links. They designed a generic framework for protecting user groups. The authors have also extended the concept to how they can protect individual users from spam attacks. The authors in [3] have given a nice analysis about why some of the profiles in twitter are suspended. They have provided statistics about various important features which can play a key role in detecting abusive user profiles. In [4] also tried to identify abusive user profiles using a set of features and trying various machine learning algorithms and comparing them.

III DATA COLLECTION

In this section we describe our data collection methodology and the different types of data we collected by crawling twitter.com website. Twitter data can be easily collected by calling the Rest API functions. Currently twitter allows 150 API calls per hour. We collected the twitter profile information of 5000 users along with their recent 200 tweets. Parts of this data (user ids) were collected from [5]. It contained both authentic and abusive user profile ids. The rest part we collected from different blogs, websites and twitter posts itself. Many abusive profiles were tagged as @spam in tweets and we manually verified them before taking them into account. The profiles were crawled and stored in XML format. After that we parsed the raw profile data to extract the meta data required for generating the features.

Another type of data we generated which was used in the profile matching part of our algorithm was a classification of user-pairs based on their relationships. We considered a portion of the twitter graph as of April, 2010 and of April, 2012. The links which did not change their characteristic were only considered. This can be a reasonable assumption that the training data for this classification is quite stabilized. We collected 400 user pairs, 100 associated with each kind of link. Four kinds of links were labeled as:

i. Bidirectional links: A follows B and B follows A.
ii. Unidirectional links: Either A follows B or B follows A.
iii. No Link with non-zero mutual friends.
iv. No link with zero mutual friends.

We used this classification to train our algorithm for profile similarity detection. Obviously the link type 1 indicates the highest similarity between profiles and so on.

IV FEATURE SELECTION AND MODELING

A. Features for Abusive User Detection

Based on analysis of twitter data that we crawled, we observed three categories of features that can targeted for automatic abusive user detection. They are:

- Profile based features
- Content based features
- Timing based features

Profile information is always a strong indicator of the general outlook of a particular user. Hence we have come up with a comprehensive list of profile based features for our classification task. They are described below:

- **Profile Description, Name, Screen name spam score:** This feature will help us measure the degree of spam worthiness of a user’s own profile description/name/screen name.

- **Follower count, Friend count, Followers to Friends ratio and Reputation of user:** These features will help us segregate abusive users from non-abusive users based on the trend we learn from the user’s social graph. User reputation score is given by,

  \[
  \text{Reputation} = \frac{(\#\text{Followers} )}{(\#\text{Followers} + \#\text{Friends} )}
  \]

- **Presence/Absence of Profile Image**

- **Profile Age of User:** This feature will help us segregate abusive users from non-abusive users based on the trend we learn from the normalized age of a user’s account.

- **Profile age:** This feature gives us an idea about how old a profile is.

- **Listed count:** This feature will help us segregate abusive users from non-abusive users based on the trend we learn from the number of groups a user subscribes to.

- **Posts/Messages per day:** We have observed that abusive users in social networks like twitter tend to post numerous posts at a go per day. This feature will help us measure this degree of differentiation.

We have also come up with a list of content based features which we believe will help us measure the abusive/non-abusive behavior of a user based on textual patterns we observe in users posts/messages. They are as follows:
- **Average tweet length**: Abusive users are seen to post shorter messages compared to the ones posted by authentic/non-abusive users.

  \[ \text{Average tweet length} = \frac{\sum \text{length of tweet}}{\text{total user tweet count}} \]

- **Average Retweets per Tweet**: Authentic users are seen to deal with more number of Retweets. This is not the normal trend with abusive users.

  \[ \text{Average Retweet per tweet} = \frac{\# \text{tweets with } "RT"}{\text{total user tweet count}} \]

- **Average mentions/replies per tweet**: Authentic users tend to mention their friends in their posts a lot more than abusive users do.

  \[ \text{Average mentions per tweet} = \frac{\# \text{posts with } @\text{friendname}}{\text{total user post count}} \]

- **Average same URL count**: Abusive users are predominantly seen to post more duplicate URLs when compared to non-abusive users.

  \[ \text{Average URL count} = \frac{\# \text{Duplicate URLs}}{\# \text{Unique URLs} \ast \text{total post count}} \]

- **Average same Hashtag count**: Abusive users are predominantly seen to post more duplicate Hashtags when compared to non-abusive users. In Twitter Hashtags are used to tag a particular topic of discussion as trending.

  \[ \text{Average unique #tags count} = \frac{\# \text{duplicate Hashtags}}{\# \text{unique Hashtags} \ast \text{total post count}} \]

- **Tweet similarity**: Abusive users are predominantly seen to post duplicate spam in bulk, grouped periodically. We believe, this feature will help us capture the degree of this behavior.

  \[ \text{Tweet Similarity} = \frac{\text{total tweet count}}{\# \text{tweet clusters}} \]

- **Average spam post count**: Abusive users are seen to have a ‘spammy’ authorship attribution when compared to authentic users. This feature will help us capture the degree of this behavior.

  \[ \text{Average spam tweet count} = \frac{\text{total spam in all posts}}{\text{total post count}} \]

- **URL per tweet, Hashtag per tweet, URL ratio, HashTag ratio**: This feature would help us capture the URL/Hashtag usage of abusive/non-abusive users. Abusive users often tend to spam a lot of their posts with URLs/Hashtags as a marketing/advertising strategy.

  \[ \text{URL per tweet} = \frac{\# \text{tweets with URLs}}{\text{total user tweet count}} \]

  \[ \text{HashTag per tweet} = \frac{\# \text{posts with Hashtags}}{\text{total user post count}} \]

  \[ \text{URL ratio} = \frac{\# \text{posts with URLs}}{\text{avg URLs per user}} \]

  \[ \text{HashTag ratio} = \frac{\# \text{posts with Hashtags}}{\text{avg Hashtags per user}} \]

We also believe, timing based attributes about users in a social network are strong differentiators in themselves. For this purpose, we have chosen two such important features. They are:

- **Average time between posts**: Abusive users are predominantly seen to make posts at a faster rate when compared to non-abusive users. This is an important observation and we believe this feature would help us capture this behavior.

  \[ \text{Average time between posts} = \frac{\sum \text{TimeStamp}(P_j) - \text{TimeStamp}(P_i)}{\text{total user post count}} \]

- **Maximum Idle duration between posts**: Abusive users are observed to be discrete in their posting behavior. They tend to post in bursts. This feature would help us capture the degree of variance of this behavior between abusive and authentic users.

  \[ \text{Average time between posts} = \frac{\text{Max} \text{TimeStamp}(P_i) - \text{TimeStamp}(P_i)}{\text{total user post count}} \]

**B. User Similarity Index**

Now that we have the model ready for abusive user detection, there can still be cases of false positives. To deal with this, we propose an automatic user similarity indexing model. This is a 4-class machine learning model based on fine-grained user similarity features between two given users. The different classes are same as mentioned in Section III. (4 categories of links) They are as follows:

- Message similarity Index
- RePost similarity Index
- HashTag similarity Index
- Description similarity Index
- Common Friend Index

The formulation for the above features is given below:

\[ a. \text{ Message similarity Index ::= } \]
### C. Social Profile Abuse Monitoring (SPAM) Algorithm

In the following figure we give a schematic of our algorithm, where user A wants to connect to user B. As mentioned earlier the SPAM algorithm helps to give a recommendation score to B so that he can decide whether or not to connect to A. It first checks profile A for abusive contents, if found it reports it to the user and its output goes to the similarity matching part. The similarity matching part takes inputs from the profiles as mentioned in section IV-B.

#### SPAM Algorithm

**Input:** Profile A and Profile B  
**Output:** i) Profile A Abusive or Authentic?  
ii) Recommended Rating to B

1. **Step 1:** Check Profile A for Abusive Behavior /* Section IV-A */
2. **Step 2:** Calculate Similarity Index between A-B /* Section IV-B*/
3. **Step 3:** Feed the Similarity Index model to the Profile classifier of Section IV-B */
4. **Step 4:** Step 3 returns a class, where it associates A-B to a particular category of link (mentioned in Section III)
5. **Step 5:** If it belongs to Category 1, output 4 stars.  
   If it belongs to Category 2, output 3 stars.  
   If it belongs to Category 3, output 2 stars.  
   If it belongs to Category 4, output 1 star.  
   Also output the result of Step 1.

### V RESULTS AND ANALYSIS

#### A. Features for Abusive User Detection

In this section we show the performance of our abusive user detection process. Also we show the performance of several top features in a Relative Frequency Histogram as shown in figure. Figure 2 and 3 shows the overall accuracy, precision and recall and F-Score for Authentic and Abusive Profiles. Figures 4 through 7 shows the performance of the top features that were used in this classification algorithm.
The red colors represent abusive users and blue authentic users. From figs. 4, we see that the number of URL’s per tweet is in the higher end for spam users. This plot also shows that a genuine user’s tweets contains at max one URL. From fig 5, we see that the number of average hashtags per tweet is in the higher end for spam users. The plot also shows that a genuine user’s tweets does not contain more than few number of hashtags in their normal behavior. From fig 6, we see that the same URL is tweeted many times by a spam user and makes this feature a strong candidate for classification. From fig 7, we see that a normal user uses a lot of mentions (@) as compared to a spam user.

B. User Similarity Index

The similarity index of two profiles is the addition of the similarity indices mention in section IV-B. The weights for each of these indices are determined by doing a 10 fold cross validation over SVM. We collected data for 4*100 links (user pairs) and labeled each connection in the following manner manually. These are the indices in the order of the weights.

- Common Friend Index
- Message similarity Index
- Description similarity Index
- HashTag similarity Index
- RePost similarity Index

VI CONCLUSION AND FUTURE WORK

In this project we have attempted to build a framework for social network privacy protection via automatic abusive user detection. We have also proposed a 4-class classification model for measuring profile similarity indexing based on fine-grained user similarity features. This would provide a measure of confidence based on which a user can filter out potential threats in the user’ social graph. Another interesting feature of our work is how we handled foreign profiles. Google API was used to translate tweets in foreign languages to english.

The future tasks will include evaluating the model on large datasets including other social networks like Facebook, Live Journal, MySpace etc. We plan to implement automatic nudity
detection from albums or pictures present in a user’s profile data using image processing techniques. Finally we hope to detect abusive and spam URLs created using shortening services like bit.ly and TinyURL by parsing the meta tags.

Data and Code we used will be updated at:
http://www.cs.sunysb.edu/~aychakrabort/courses/cse508/

REFERENCES


