Detection Of Spammers In Twitter Applications

1Maithili. K and 2Murugappan. S

1Research Scholar,Computer Science and Engineering,Manonmaniam Sundaranar University, India. 
2Department of Computer Sciences, TamilNadu Open University, India.

Received 7 June 2016; Accepted 12 October 2016; Available 20 October 2016

ABSTRACT

Social Networks plays a very important role to communicate and interact with one another. At the same time, a variety of prohibited behaviors that violate all social networks. Each Social Network site uses different approaches to spread spam. Especially on twitter, the spam is manifested by tweets, fake profiles, hashtags chats, URLs, following a large number of users, and links, etc. In this aspect, we focus on micro-blogging social networking site Twitter for spam detection. Twitter is the most known social network in which the spammers can easily spread the spam tweets by using the popular hashtags those trends on the twitter. They can also mention the users without the permission of that victim. They don’t have to be a follower or friend to mention them on the spam tweet. For identifying such spammers, we propose a new framework for detection of spammers in Twitter applications using supervised learning methods. In this paper, first, we collect a dataset from Twitter using Twitter API. Second, collected dataset is converted to comma separated values. Third, pre-processing the data to clean the dataset. Fourth, a minimal set of features like user based features and content based features are extracted, and labeled based on the spam words that contains from the preprocessed datasets. Finally, three classification algorithms applied, namely Naive Bayes, Random Forest and NBTree. The performance of the system is evaluated with confusion matrix, precision, recall, and F-measure in 5-fold and 10-fold cross validation.

KEYWORDS: 5-fold and 10-fold cross validation, Naive Bayes, NBTree, Random Forest, Spam, Twitter.

INTRODUCTION

Spam refers to an online communication and interaction, which is sent to the user without their permission. detection in twitter refers to a variety of prohibited behaviors that violate the twitter rules. This can be generally described as unsolicited, repeated actions that negatively impact other users in twitter. It includes many forms of automated account interactions, follows and behaviors as well as attempts to mislead or deceive users for different circumstances. To comprehend the way spammers work in the Twitter community, it is good to bring to mind the salient points of how social media behaves, especially Twitter. Tweeters communicate through short messages (140 characters max), that can be viewed by the other tweeters “followers” and can be “re-tweeted”. Twitter accommodates two types of relationships: “followers” and “following”. Once a person creates a profile in twitter he can go establishing relationships in twitter with other members of twitter by becoming a “follower”. He will then receive all the tweets posted by the person whom he is following. Similarly other members of twitter can establish a relationship with him by becoming a “follower of him”. This relationship need not be symmetric, that is if Mr. A is a follower of Mr. B, Mr. B need not be a follower of Mr. A.

Twitter also has the option where a user can post messages or status updates to another user. It is this functionality which miscreants can abuse and spam or flood another user's page with useless, junk or spite messages without permission [12].

To Cite This Article: Maithili. K and Murugappan. S., Detection Of Spammers In Twitter Applications. Advances in Natural and Applied Sciences 10(14): Pages 140-147
followers’ profile, in order to misguide them to phishing pages or to spread malware. This property of Twitter which does not require relationships to be symmetric is the root of the problem behind these malicious spamming. In particular, the behaviors constitute spamming on Twitter. The following list of examples constitutes spamming in the twitter rules. Some common tactics that occurs in spam accounts such as posting harmful links (including links to phishing or malware sites), aggressive following behavior (mass following and mass unfollowing for attention), abusing the reply or mention functions to post unwanted messages to users, creating multiple accounts (either manually or using automated tools), posting repeatedly to trending topics to try to grab attention, repeatedly posting duplicate updates, posting links with unrelated tweets[10]. Social spam refers to unwanted messages or contents appearing on social networks and any website with user-generated content. It can be manifested in many ways, including bulk messages, profanity, insults, hate speech, malicious links, fraudulent reviews, fake friends, and personally identifiable information[11]. Twitter is a most popular micro-blogging social networking site among people because of its simple working mechanism. Each Social Network site uses different approaches to spread spam. Especially on twitter, the spam is manifested by tweets, fake profiles, hashtags chats, URLs, following a large number of users, and links, etc. In this paper, we first extracted the twitter id, and the tweet text from the collected dataset by using JSON module. Second, the preprocessing techniques can be applied for tokenization, sentence splitter, stemming and removal of stop words, punctuations, quotations and urls for cleaning the collected tweets. Third, the features can be extracted from the cleaned data. Fourth, supervised learning methods (Random Forest, Naive Bayes, and NBTree) can be applied for the extracted features. Fifth, the performance of the system can be estimated based on the system accuracy with precision, recall and f-measure.

II. Literature Survey:
Claudia Meda et al., [1] proposed an approach that makes a comparative study of the research shows that Machine Learning (ML) may provide a powerful tool to support spammer detection in Twitter. The present system compares the performance of three different machine learning algorithms in tackling this task. The experimental session involves a publicly available dataset. Zachary Miller et al., [2] proposed the spam detection as a classification problem using data streams. The author first viewed the spam detection as an anomaly detection problem. Second, the authors were introduced 95 one-gram features from the tweet text alongside the user information analyzed in previous studies. Finally, two stream clustering algorithms were modified namely StreamKM++ and DenStream to facilitate spam identification to effectively handle the streaming nature of tweets. Both the algorithms clustered normal Twitter users, treating outliers as spammers. StreamKM++ and DenStream approaches achieved 97.1% accuracy and 84.2% F-Measure and 94.0% accuracy and 74.8% F-Measure respectively. The authors reported that these findings suggest the addition of one-gram features enhances spam detection. Although these algorithms independently demonstrated good detection. Chi-Yao Tseng et al., [3] proposed the mail detection system MailNet devised with incremental Support Vector Machine (SVM).

In SVM, system is not robust in diverse environments, and no update scheme is provided to catch the feature changes of evolving networks. It also provides an incremental update scheme to efficiently re-train an SVM model, and a live data set from a university-scale email server. This show that the proposed model is efficient and effective, thus applicable to the real world. Sangho Lee et al., [4] proposed an approach for suspicious URL detection. They have collected a large number of tweets from the Twitter public timeline and trained a statistical classifier with features derived from correlated URLs and tweet context information. The classifier has high accuracy and low false-positive and false negative rates. It also presents WARNINGBIRD as a real time system for classifying suspicious URLs in the Twitter stream. Xin Jin et al., [5] proposed a scalable and online social media spam detection system for social network security. The authors employed GAD clustering algorithm for large scale clustering and integrate it with the designed active learning algorithm to deal with the scalability and real-time detection challenges. They choose popular pages with over 500,000 fans (an average of 2 million fans for each page) as basic sensors to monitor the public posting activities in the social network, and setup a website to show the recently detected spam messages/photos and the corresponding spammers or infected users. In this way, the audience can obtain an intuitive understanding about the essence of online social spam detection for social media.

III. Proposed System:
The proposed system follows the basic steps similar to the existing system like the data collection, preprocessing, feature extraction, classification and the spam analysis. In the existing system, the most recently tweeted twenty tweets are considered for specific users in the analysis. In the proposed system, the random tweets for the multiple users are considered, and extracted the minimal set of features from the preprocessed dataset. Then three different supervised learning methods are used for the spam analysis. The objective of the proposed system is to classify the spams and hams from the collected dataset. The data collection process is carried out by using the twitter API. Tweepy is a python wrapper for the twitter API that is used to collect the
tweets in the JSON format. Also, the python JSON module used to retrieve the objects that want to use in this system. The main scope of this work is to construct a framework for multiple users in twitter spam detection and that framework will collect the raw tweets using the Twitter API to create a dataset. It also constructs semi-labels for cleaned tweets as spam and ham and this framework improves the performance of the Random Forest algorithm than other supervised learning methods.

3.1. Data Collection Process:

Data collection is the process of collecting the tweets from the twitter stream. In twitter, the Application Programming Interface (API) is provided that can be used by the others[13]. The input is represented by Twitter messages, which form the training set for the classification task. Tweets are collected using Twitter Streaming API [8]; this process is necessary in order to obtain a large amount of messages for the classification task. The tweepy framework is used to collect the twitter feeds. Tweepy is a twitter API wrapper which was written in python. By the help of twitter API the tweets are collected in the format of JSON. After that the English language tweets are separated from the original raw tweets[12]. JSON is a lite weight format that is used for the communication over the web applications. By using the python JSON module, the objects are separated, and also the twitter id and the tweet text are separated from the JSON file for the preprocessing phase.

3.2. Preprocessing:

Data preprocessing is an important step in the data mining process. The phrase garbage in-garbage out is particularly applicable to data mining. Data pre-processing includes cleaning, normalization. The final product is the training data set[12]. The preprocessing phase contains the following methods to clean the collected dataset.

3.2.1. Tokenization:

Tokenization is the task of chopping the whole text up into pieces called tokens at the same time throwing away certain characters, such as punctuations, special characters and non ASCII characters.

3.2.2. Sentence Splitter:

Sentence splitting is done with the help of the tokens when passed a list of tokens. Each sentence should be represented as a list of tokens, so the function as a whole returns a list of lists of tokens.
3.2.3. Stemming:
Stemming is the process of reducing inflectional forms and sometimes derivationally related forms of a word to a common base form. E.g. - am, are, has been converted as ‘be’.

3.2.4. Removal of Stop Words:
Stop words are words which are filtered out before or after processing of data. Stop words can cause problems when searching for phrases.

3.2.5. Word Net:
Word Net is an English language lexical database. The lexical database groups English words into sets of synonyms called synsets.

3.3. Feature Extraction:
Feature extraction involves reducing the amount of resources required to describe a large set of data. Performing analysis of complex data is one of the major problems due to the large number of variables involved. The analysis requires large amounts of memory and computation power to fit with the training model. Feature extraction is a method used to construct derived features from an initial set of measured data with sufficient accuracy. In [1], the idea is proposed to a set of 62 features for detecting spammers. In this paper, only the minimal numbers of features are extracted from the collected and preprocessed tweets. It is shown in the Table 1.

3.3.1. User Based Features:
User based features employed in this thesis have been listed below:

- **Profile Details:**
  Details about the twitter users name, age, location and language are considered in this proposed system.

- **Number of Followers/Friends:**
  Followers are the person who can get their tweets immediately after they post it on their account. It is about the total number of followers in the twitter account.

- **Number of Followings, Followers/Following Ratio:**
  Twitter users can follow others to get their tweets on their walls. This is about the ratio of number of following and followers.

- **Reputation:**
  Reputation can be calculated by their following/follower ratio. Their followers should be higher than the following count.

- **Age of Account:**
  It is about how long their account is being active from the account creation.

- **Average Time between Tweets Posting Time Behavior:**
  It is the difference between two tweets time interval. The time interval should be very less compared to the valid users.

- **Tweet/post Frequency:**
  This is the frequency of how often they tweet in their account. Legitimate users can tweet less frequently than the spammers.

3.3.2. Content Based Features:
A content based feature employed in this thesis has been listed below:

- **Number of Hashtags(#):**
  It is about the count of hashtags they used in their tweet. Hashtags are used to filter the topics easily. Spammers have the behavior to tweet an unwanted promotions or tweet by misusing this mechanism. They post their tweet with the popular trending topics. So it can be easily reach by the other peoples.

- **Number of URLs in Tweets:**
  The count of URLs contained in the tweet is calculated. The destination cannot be revealed until we click that link.
Mentions (@):
@ symbol is used to mention the friends or any other person directly in the tweets. There is no restriction in using this symbol. Anyone can also mention on their tweet without the user permission. This can also lead to spread the spam.

Retweets:
Retweeting is a method to share one person post to others. It is similar to sharing option in facebook. Popular tweets can be retweeted many times.

Spam Words:
Tweets may contain the spam words. The number of spam words in each tweets is counted. The spam words used in this system is collected from the spam assassin page [9].

Trending Topics:
Trend topics are the popular topics that are being hashtagged and posted in twitter. This can be used by the spammers to spread the spam.

Table 1: Feature Extraction Details

<table>
<thead>
<tr>
<th>URLs</th>
<th>ftw_{URL} : \left( \sum_{i=1}^{N} tw_{URL(i)} \right)</th>
<th>tw_{URL} : The number of tweets with at least an URL inside</th>
</tr>
</thead>
<tbody>
<tr>
<td>Replies</td>
<td>f_{URL} : \left( \sum_{i=1}^{N} URL_{i} \right)</td>
<td>URL_{i} : the number of URLs on each tweet</td>
</tr>
<tr>
<td>Spam words</td>
<td>f_{S} : \left( \sum_{i=1}^{N} tw_{S(i)} \right)</td>
<td>tw_{S(i)} : the number of tweets with at least an @ inside.</td>
</tr>
</tbody>
</table>

Fractions all of features

<table>
<thead>
<tr>
<th>ftw_{URL}</th>
<th>\frac{ftw_{URL}}{N}</th>
<th>Fraction of tweets with URLs</th>
</tr>
</thead>
<tbody>
<tr>
<td>f_{R}</td>
<td>\frac{f_{R}}{N}</td>
<td>Fraction of tweets replied</td>
</tr>
<tr>
<td>f_{S}</td>
<td>\frac{f_{S}}{N}</td>
<td>Fraction of tweets with spam words</td>
</tr>
<tr>
<td>f_{URL_{i}}</td>
<td>\frac{f_{URL_{i}}}{N}</td>
<td>Number of URLs on each tweet (mean):</td>
</tr>
<tr>
<td>f_{T}</td>
<td>\frac{f_{T}}{N}</td>
<td>Time between posts (mean)</td>
</tr>
</tbody>
</table>

3.4. Classification Algorithms:
The extracted features are classified using the following algorithms:

3.4.1. Naive Bayes Algorithm:
A Naive Bayes classifier assumes that the value of a particular feature is unrelated to the presence or absence of any other feature, given the class variable. An advantage of naive bayes is that it only requires a small amount of training data to estimate the parameters necessary for classification.

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}$$  \hspace{1cm} (3.1)

$$P(H | E) = \frac{P(H \land E)}{P(E)}$$  \hspace{1cm} (3.2)
\[
P(E|H) = \frac{P(H \land E)}{P(H)} \quad (3.3) \\
P(H \land E) = P(E|H)P(H) \quad (3.4) \\
P(H|E) = \frac{P(E|H)P(H)}{P(E)} \quad (3.5)
\]

An input and output detail for these classifier has been listed below

**Input:**
- Features \(d\)
- A fixed set of classes \(C = \{c_1, c_2, \ldots, c_j\}\)
- A training set of \(m\) hand-labeled documents \((d_1, c_1), \ldots, (d_m, c_m)\)

**Output:**
- A learned classifier \(y: d \rightarrow c\)

### 3.4.2. Random Forest Algorithm:

It is an ensemble learning method for classification and regression that operates by constructing a certain number of decision trees at training time and outputting the class selected with a majority vote by individual trees. The method combines the general technique of bootstrap aggregating, or bagging, with the random selection of features [1].

Pseudo code
For \(nt = 1\) to \(NTREES\)
    Choose \(d\) features from \(D\) without replacement;
    Sample, with replacement, \(n\) training patterns from \(X, Y\); call these \(Xnt, Ynt\);
    Train a decision tree \(fnt\) on \(Xnt, Ynt\).
End

**IV. Experimental Results And Performance Evaluation:**

The implementation of this framework is done in the python language. Different python modules are written to carry out the data collection, preprocessing, feature extraction tasks. And after that WEKA [6] tool is used for the classification phase for the spam analysis tasks. WEKA is a classification tool that used for the data mining tasks. It contains all the classification algorithms, and trees that can be applied for the data analysis [6]. It also gives the functionality for the visualization of analyzed data. Python is a widely used language for natural language processing. Its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C or Java. It provides a huge library support to efficiently process natural language text such as packages for manipulating regular expressions. It also allows us to perform computations on a collection such as lists, dictionary and set in a very fast manner. Python 2.7 [14] was used in the project. First, Data collection process is carried out using the `twitter_streaming.py` module. Second, The collected dataset is converted to comma separated values by using the `json2csv.py` module. Third, Preprocessing task is done by using the `preprocessing.py` module. This will clean the dataset. Forth, the features are extracted and labeled based on the spam words that contains from the preprocessed datasets as shown in Figure 2.

![ARFF Input File for Classification](image)

**Fig. 2:** ARFF Input File for Classification
The experimentation is carried out by using the 15844 tweets of different users in python environment. In this tweets, only 844 labeled spams and 15000 labeled hams are considered as shown in the table 2. The extracted features are urls, hashtags, mentions, and spam words counts generated as training dataset in the arff format. This training data is applied into WEKA tool[6] for classification. The confusion matrix for spam classification is shown in Table 3.

**Table 2: Labeled Datasets for Spam/Ham**

<table>
<thead>
<tr>
<th></th>
<th>Spam</th>
<th>Ham</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>844</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>15000</td>
<td></td>
</tr>
</tbody>
</table>

**Table 3: Confusion Matrix for Spam Classification**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>tpA</td>
<td>eAB</td>
</tr>
<tr>
<td>eBA</td>
<td>tpB</td>
</tr>
</tbody>
</table>

Where tpA and tpB are true positives and eAB and eBA are true negatives. The following metrics are used to measure the performance of system for identifying spammers in twitter applications. Table 4 shows the 5 fold cross validation results for the Naive Bayes, Random Forest Algorithm and NBTree and Table 5 shows the 10 fold cross validation results for the Naive Bayes, Random Forest Algorithm and NBTree algorithms.

**Precision:**

Precision is the portion of true positive predicted instances against all Predicted instances. P represents precision and tp for true positives and fp represents false positives.

\[ P = \frac{tp}{tp + fp} \]  

**Recall:**

Recall is the portion of true positive predicted instances against all actual positive instances. R represents recall, fn represents for false negatives.

\[ R = \frac{tp}{tp + fn} \]  

**F-Measure:**

F measure is a harmonic average of precision and recall. P represents precision and R represents recall.

\[ F = 2 \frac{P \cdot R}{P + R} \]  

**Table 4: 5-fold Cross Validation across Classifiers**

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Confusion matrix</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Forest</td>
<td>130 14696</td>
<td>0.52</td>
<td>0.98</td>
<td>0.97</td>
<td>94.83</td>
</tr>
<tr>
<td>Naïve Bayes</td>
<td>348 14333</td>
<td>0.34</td>
<td>0.95</td>
<td>0.96</td>
<td>92.65</td>
</tr>
<tr>
<td>NBTree</td>
<td>350 14602</td>
<td>0.46</td>
<td>0.97</td>
<td>0.97</td>
<td>94.37</td>
</tr>
</tbody>
</table>

**Table 5: 10-fold Cross Validation across Classifiers**

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Confusion matrix</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Forest</td>
<td>330 14696</td>
<td>0.52</td>
<td>0.97</td>
<td>0.97</td>
<td>94.83</td>
</tr>
<tr>
<td>Naïve Bayes</td>
<td>348 14329</td>
<td>0.34</td>
<td>0.95</td>
<td>0.96</td>
<td>92.63</td>
</tr>
<tr>
<td>NBTree</td>
<td>352 14593</td>
<td>0.46</td>
<td>0.97</td>
<td>0.97</td>
<td>94.32</td>
</tr>
</tbody>
</table>
criminal behavior make processing the data, extracting a measure. In overall, the Random Forest algorithm outperforms Based Spam Detection System For Social Media: 12).

check the observations for a larger sample size data.

Fig. 3: 5-fold Cross Validation across Classifiers

Figure 3 shows the results for the 5 fold cross validation across the Naive Bayes, Random Forest Algorithm and NBTree and Figure 4 shows the results for the 10 fold cross validation across the Naive Bayes, Random Forest Algorithm and NBTree algorithms. The results of the random forest, naive bayes, and support vector machine are performed using 5 fold and 10 fold cross validation as shown in the table 4 & 5. The Random Forest algorithm achieves 94.83% accuracy with 0.74% precision, 0.68% recall, and 0.70% f-measure. Naive Bayes algorithm achieves 92.63 accuracy with 0.65% precision, 0.68% recall and 0.68 F-measure. NBTree algorithm achieves 94.32 accuracy with 0.76% precision, 0.69% recall and 0.69.

V. Conclusion And Future Work:
The proposed framework is introduced based on supervised learning method for the detection of spammers on Twitter. The cyber-criminal behavior makes the challenging task to identify malicious users. The research in the system is implemented in various steps like, data collection process, pre-processing the data, extracting a minimal set of features for user and content, labeling the features as spam and non-spam, and applying classification algorithms. The Random Forest, Naive Bayes and NBTree algorithms are used spam analysis in 5-fold and 10-fold cross validation. In 5-fold cross validation, the algorithms achieves the accuracy of 94.83%, 92.65%, and 94.37% respectively. The Random Forest algorithm outperforms the system accuracy (84.83%) with 0.74% precision, 0.68% recall, and 0.70% f-measure. In 10-fold cross validation, the algorithms achieves 94.83%, 92.63, and 94.32% respectively. The Random Forest algorithms achieves the accuracy of 94.83% with 0.74% precision, 0.68% recall, and 0.71% f-measure. In overall, the Random Forest algorithm outperforms 94.83% accuracy with 0.74% precision, 0.68% recall, and 0.70% f-measure. In future, the proposed system can be carried out for different online social networking sites, contents, e-mails. Also, to create a web crawler to gather twitter details and re-run the algorithm and re-check the observations for a larger sample size data.

REFERENCES

8. Twitter Streaming API, Https://Dev.Twitter.Com/Docs/Api/Streaming