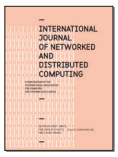
International Journal of Networked and Distributed Computing
Vol. 7(1); December (2018), *pp.* 43–50
DOI: 10.2991/ijndc.2018.7.1.6; ISSN 2211-7946
https://www.atlantis-press.com/journals/ijndc



Blacklist Creation for Detecting Fake Accounts on Twitter

Myo Myo Swe*, Nyein Nyein Myo

University of Computer Studies, Web Mining Lab, Mandalay Patheingyi Mandalay, Yangon 1001, Myanmar

ARTICLE INFO

Article History

Received 5 October 2018
Accepted 13 November 2018

Keywords

Topic extraction approach
keyword extraction approach
DECORATE

ABSTRACT

Social networking sites such as Twitter, Facebook, Weibo, etc. are extremely mainstream today. Also, the greater part of the malicious users utilize these sites to persuade legitimate users for different purposes, for example, to promote their products item, to enter their spam links, to stigmatize other persons, etc. An ever increasing number of users utilize these social networking sites and fake accounts on these destinations are turned into a major issue. In this paper, fake accounts are detected using blacklist instead of traditional spam words list. Blacklist is created using topic modeling approach and keyword extraction approach. We evaluate our blacklist based approach on 1KS-10KN dataset and Social Honeypot dataset and compared the accuracy with the traditional spam words list based approach. Diverse ensemble creation by oppositional relabeling of artificial training examples, a meta-learner classifier is applied for classifying fake accounts on Twitter from legitimate accounts. Our approach achieves 95.4% accuracy and true positive rate is 0.95.

1. INTRODUCTION

People use Twitter to share their feelings, news, events and to post their daily activities such as eating, drinking, travelling, etc. Therefore, malicious users can check everyone's activities from their timeline and Twitter becomes a place for malicious users to commit the crimes. These malicious users create fake accounts and spread various fake news, links and photos. Most of the internet users are not awareness of these fake accounts; they accepted their requests and became victims for these fake users. Therefore, fake accounts detection on Twitter is necessary for everyone who uses the social networking sites.

Twitter is a free social networking site and allows a user to post 280 characters to express their feelings and thoughts. The simplicity of sharing and getting to client created content, including sentiments, news, and drifting subjects are the esteem of the Twitter. Twitter gives a chance to produce substantial movement and income, particularly since it has a huge number of clients. These open doors make Twitter a key objective of spammers. It is simple for people to recognize spammers from genuine clients, however the presence of spammers squanders users' time and consideration, puts clients in danger in getting to malignant and hazardous substance, and cheapens Twitter's administrations and the general online informal community.

The fame of Twitter has prompted the ascent of undesirable, troublesome data from social spammers. Fake users on Twitter are remarked as malicious users who endeavor to increase social impact and create abnormal activities which adversely effect on authentic users. Malicious users use shortened links and shortened Uniform Resource Locator (URL) and they also express their ideas with repetitive words to grab the real users' attentions. Fake users have their fake patterns and can be easily check them by carefully examining their tweeting patterns. Fake users also used fake words in their tweets and previous researchers are detected these fake words using traditional spam words list. In these spam words lists, spam words are listed based on the traditional spam email, but social networking site is very different from traditional spam email. Texts in social networking sites are not formal and fake users post various kinds of fake news such as politics, sports, weather, and celebrity, etc. But most of the spam email words are about advertising, commercial, and marketing. Therefore, the most previous fake accounts detection using these spam words lists are not achieved the best accuracy and false positive rates are high.

Many former researchers detected fake accounts on Twitter using various features. Some of the researchers detected fake accounts based on content of the tweets. Some researchers detected these fake accounts based on users' profiles. Other researchers detected using both content and profile features. Some of the Twitter users show their profile details and their tweets, but some of the others do not show. Crawling tools can extract the publicly available data from users' timeline. Most of the users show their tweets to public, but do not explicitly describe the profile detail. We cannot get the profile detail of the user easily. Therefore, in our detection approach, tweets and features are extracted based on the most recent 20 tweets. In this paper, the aim of the approach is to detect fake account on Twitter based on the content of tweet. The major contribution of this approach is to create a blacklist that can effectively extract the fake features from the fake accounts.

* *Corresponding author. Email: myomyoswe.maeu@gmail.com*

This paper is composed as follows. In Section 2, the relevant previous work on fake account detection on Twitter is presented. Section 3 describes the detail description for the creation of blacklist to detect fake accounts. The architecture of the fake account detection system is presented in Section 4. Evaluation of the proposed approach is discussed in Section 5. Finally, the paper concludes in Section 6.

2. RELATED WORK

Some of the fake users were bots and they use automatic twitting tools and this type of fake users can be found by checking their tweeting patterns. Chu et al. [1] created three components; first component was utilized for entropy calculation, second component was used for calculating spam probability, and third component was utilized to compute the account properties statistics. These three components were based on users' tweet content and users' profiles. Bots post tweets periodic and regular timing. They have automation behavior and this behavior could be found using the entropy component. Spammers write spam contents and these spam words could be found using spam detection component. Spam detection component calculated the spam probability using traditional Naïve Bayesian. The account properties statistics such as URL ratio, tweeting device makeup, followers to friend's ratio were extracted from the user log by the account properties component and these statistics were send to the decision marker. Random Forest classifier was used for decision marker. The decision marker classified the account into three classes: human, cyborg or bot.

To create the directed social graph model, Wang [2] used the followers and friends relationship. Two graph-based features such as number of following, number of followers and reputation and four content-based features such as number of tweet similarity, number of mentions, number of URLs and number of hashtags were extracted from the user profile and user content. The author used four machine learning classifiers such as Support Vector Machine, Decision Tree, Naïve Bayes and Neural Network to classify the spammers. Naïve Bayes was the best classification algorithm in this system. This approach achieved 93.5% accuracy. Reputation feature was the most effective feature in this approach.

Liu et al. [3] proposed two new features based on the topic modeling approach. They extracted global outlier standard score and local outlier standard score from users' tweets. The topic probability vector was achieved using the latent Dirichlet allocation. They compared their approach with three baseline approaches. Their approach achieved better accuracy than the baseline approach.

Benevenuto et al. [4] recognized spammers using tweet content and user behavior attributes. They created own dataset for spammers detection. The author used Support Vector Machine classifier and this classifier achieved 87.4% accuracy. To detect spam profiles on Twitter, McCord et al. [5] extracted features from users' tweets and users' profiles. The authors applied four classifiers such as Naïve Bayesian, *K*-nearest neighbor, Support Vector Machine and Random Forest. The detection accuracy of this approach got 95.7%.

Two spammer detection approaches: user centric approach and URL centric approach were proposed by Amit et al. [6]. In this approach, the author proposed 15 new features and combined with existing features to detect fake accounts. Their approach was tested on small dataset. Yang et al. [7] used 10 new features including three graph-based features, three neighbor-based features, three automation-based features and one timing-based features. Their new features were robust for evasion, but these could not be easily extracted.

Meda et al. [8] recognized spammers on Twitter using 13 features. The author applied Random Forest, Decision Tree, AdaBoost, Bagging and LogitBoost classifiers to detect spammers. In this approach, Random Forest achieved the best detection result. Chakraborty et al. checked the links whether these were harmful URLs, porn URLs or not. This was approached in two steps [9]. The authors used twenty features for classification. Four machine learning classifiers were applied and Support Vector Machine gave the best accuracy.

Lee et al. created Social Honeypots comprising of real profiles that distinguished suspicious clients and its bot gathered proof of the spam by creeping the profile of the client sending the undesirable companion solicitations and URLs in MySpace and Twitter. They used highlights of profiles as features such as users' posting conduct, substance and companion data to build up a classifier that have been utilized for recognizing spammers. After investigation, profiles of clients who sent spontaneous companion solicitation to these Social Honeypots in MySpace and Twitter have been gathered. The authors applied A Library for Support Vector Machines (LIBSVM) classifier for classification. The advantage of this approach is that it has been approved on two unique mixes of dataset – first dataset contains 10% spammers and 90% non-spammers and second dataset contains 10% non-spammers and 90% spammers [10].

Twitter encourages its users to show spam users to them by making "@spam" message. This feature was applied to detect spam profiles by Gee et al. They collected real users with Twitter API and malicious users from "@spam" message. The collected data were saved in, comma-separated values (CSV) file for classification. Their approach utilized Naïve Bayesian classifier and achieved 89.3% accuracy. Features used in their approach were not technical, therefore it achieved less precision [11].

To identify long-surviving spam accounts on Twitter, Lin et al. utilized two features: URL rate and interaction rate. Former researchers have utilized numerous attributes for classification of fake accounts such as number of followers, number of followings, number of favorites, number of tweets, number of hashtags, number of mentions, number of hyperlinks, reputation, etc., but these approaches are not given the best accuracy. Therefore, the author proposed two effective features: namely URL rate and interaction rate for classification. Twitter API was used for crawling 26,758 accounts and 816 long surviving accounts. To classify the accounts into normal and long-surviving account, J48 classifier was applied for classification. 86% precision was achieved. Restriction of this method was that using two features was not the best, because malicious users could keep low URL rate and low interaction rate [12].

The previous works mentioned above used content-based, profile-based and network-based features for detecting fake accounts on Twitter. Because of data crawling difficulties, profile-based and network-based features cannot extract easily, only content-based features are used in our work. Content-based features such as fake words count, fake words ratio in previous works are extracted from the traditional spam words list and these systems did not get the best accuracy. Therefore, in our approach, we create a blacklist for

content-based features extraction and the best accuracy is achieved using our blacklist.

3. BLACKLIST CREATION

This section presents the creation of blacklist. The system flow of blacklist creation is shown in Figure 1. The blacklist creation includes four steps: (1) data collection, (2) preprocessing, (3) topic extraction and (4) keyword extraction. The detailed description of these four steps are as follows.

3.1. Data Collection

1KS-10KN dataset has two sub-datasets; real dataset and fake dataset. Fake dataset contains 1000 identified malicious Twitter accounts. Real dataset contains 10,000 normal Twitter accounts. At first, only the fake tweets are extracted from the fake dataset. To create the blacklist, 1,45,095 tweets of 1000 fake users from this fake dataset are used. In this step, tweets that are not written with English language are excluded from this dataset. Links and URLs (http:// (or) https:// (or) www.) are also excluded from the tweets. Mention (@username) and hashtag (#) are also excluded from the tweets.

3.2. Preprocessing

In preprocessing step, the collected tweets are preprocessed. At first, we perform tokenization on the tweets. In tokenization, tweets are splits into little pieces or token. Bigger content of tweets can be tokenized into sentences; sentences can be tokenized into words, etc. Tokenization is additionally called to as content division or lexical examination. Once in a while division is utilized to allude to the breakdown of an extensive lump of content into pieces bigger than words (e.g. sections or sentences), while tokenization is saved for the breakdown procedure which results only in words. After tokenization, to discover the root or stem of a word, stemming algorithm is utilized. Porter stemming algorithm is applied to that tweets. Stemming is a work that removes morphological and inflexional endings of words, for examples, foxes to fox, and seller, selling to sell. And then, we perform lemmatization. Lemmatization is identified with stemming; varying in that lemmatization can catch authoritative structures in light of a word's lemma. For instance, stemming "better" would neglect to restore its reference shape (another word for lemma); be that as it may, lemmatization would result in the accompanying: better to good. It ought to be anything but difficult to perceive any reason why the execution of a stemmer would be the less troublesome accomplishment of the two. After lemmatization, all characters are changed to lowercase. Numbers in tweets are removed (or convert numbers to literary portrayals). Punctuation are also removed (for the most part some portion of tokenization, yet at the same time worth remembering at this stage, even as affirmation) and white space are stripped. Default stop words (general English stop words) are removed. Stop words are dialect particular practical words, are visit words that convey no data (i.e., pronouns, relational words, conjunctions). Stop words such as a, an, the, of, containing in the collected tweets are removed. Because most of the stop words are frequently occur in text and they are not useful.

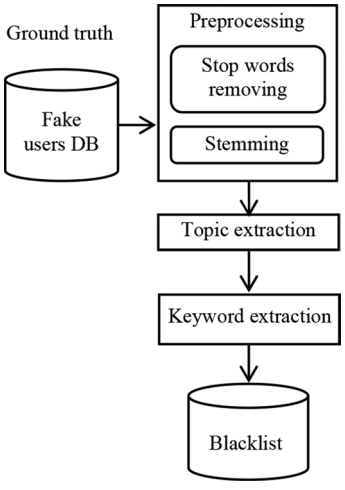


Figure 1 | Blacklist creation system flow

3.3. Topic Extraction

Latent Dirichlet allocation is used for topic extraction. Latent Dirichlet allocation is proposed by Blei et al. [13] as model for extracting topics. It is a three-level various leveled Bayesian model, in which everything of an accumulation is displayed as a limited blend over a fundamental arrangement of subjects. Each topic is, in turn, modeled as an infinite mixture over an underlying set of topic probabilities. In the context of text modeling, the topic probabilities provide an explicit representation of a document. Each document is seemed as a bag of words $W = \{wi_1, wi_2, \ldots, wi_M\}$ and M is the number of words. Each word is distributed to one of the document's topics $Z = \{zi_1, zi_2, \ldots, zi_K\}$ and K is the number of topics. Φm is a multinomial distribution over words for topic m. θi is an another multinomial distribution over topics for document i. σ is the parameter of the Dirichlet prior on the per-document topic distributions. β is the parameter of the Dirichlet prior on the per-topic word distribution. We set σ, β, and M to 0.1, 0.01 and 10. The entire content of each Twitter user is marked as one document. Gibbs sampling is used to speed up the inference of Latent Dirichlet Allocation (LDA). The used dataset is $X = [X_1, X_2, \ldots, X_k] \in R_k{}^*M$, where k is the number of users. Each element $X_i = [p(z_1)\, p(z_2) \ldots p(zM)] \in R_1{}^*{}_M$ is a topic probability vector for the i^{th} document. The number of top words in our system is 50. After applying LDA, $Y = [Y_1, Y_2, \ldots, Y_{10}]$ and $Y_i = [p(w_1)\, p(w_2) \ldots p(w_{50})]$; $p(w_1) > p(w_2) > \ldots > p(w_{50})$ are achieved. Each Y_i is regarded as one document and document set Y is used as input corpus for the next step.

3.4. Keyword Extraction

TF–IDF stands for term frequency–inverse document frequency, and the TF-IDF weight is a weight often used in information retrieval and text mining. In our approach, TF–IDF is applied for extracting keywords. TF-IDF or TF–IDF is a numerical statistic method. TF–IDF is used to prove the importance of a word in a document or corpus [14]. It is used as a weighting factor in searches of information retrieval, text mining, and user modeling.

The TF–IDF value increases proportionally to the number of times a word appears in the document and is offset by the frequency of the word in the corpus which helps to adjust for the fact that some words appear more frequently in general. The new corpus, $Y = [Y_1, Y_2, \ldots, Y_{10}]$ that is achieved from previous step is inserted as input in the TF–IDF algorithm. TF–IDF is utilized to rank each word of the documents. We choose the top 50 words of each document and that are marked as fake words and these fake words are inserted into blacklist to create the blacklist corpus. The blacklist contains 500 fake words and we use this blacklist for detecting fake accounts.

4. FAKE ACCOUNTS DETECTION ON TWITTER

In this section, fake accounts are detected using content-based features. Content-based features are extracted from text of the tweets posted by the user. After features extraction, Diverse Ensemble Creation by Oppositional Relabeling of Artificial Training Examples (DECORATE) classifier is applied on the extracted features to classify fake or real users. The system flow of the fake account detection on Twitter is shown in Figure 2. Fake accounts detection is a three-step process. First, the content of the user that we want to classify are crawled using the Twitter API. And then, features are extracted from the content of tweets. Finally, we classify the account of the user using DECORATE classifier. The detailed descriptions for the steps of the fake account detection are mentioned below.

4.1. Features Extraction

To classify fake accounts on Twitter, we first extract the features that can distinguish fake accounts from legitimate accounts. Most of the researchers detected fake accounts based on users' profile, content and network. But most of the Twitter users do not show their profile detail and their followers and followings relationships, they show only their tweets content to public. Profile-based features and network-based features extraction can cause extra cost and time. Therefore, in our approach, we extracted features from the content of the user which are publicly available. About 14 content-based features are extracted for detecting fake accounts. Number of fake words and fake word ratio features are extracted using blacklist rather than traditional spam words list. About 14 content-based features are number of tweets, number of URLs, number of hashtags, number of mentions, number of retweets, number of fake words, fake words ratio, URLs ratio, hashtags ratio, mention ratio, total number of words, mean time between tweets, standard deviation time between tweets and extreme idle duration time between tweets. These 14 content-based features are briefly described as follows.

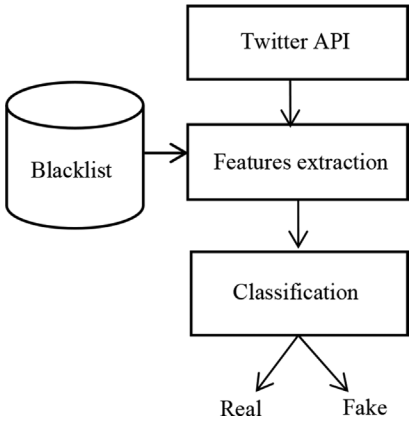


Figure 2 | Fake account detection system

4.1.1. Number of tweets

Malicious users present more tweets to be more dynamic and more eager to connect with others. They posted tweets in a particular time interim utilizing specific mechanized tweeting instruments and programming, for example, Twitter API and auto twitting device. In this manner, number of tweets is a property for recognition.

4.1.2. Number of URLs

Users are enabled to post tweets with 280 characters so the greater part of the malicious users utilize abbreviated URLs to post their tweets. Twitter cannot check these URLs and phony users likewise use these URLs that are prompting the noxious page by clicking these URLs. In this manner, number of URLs containing in tweets is the noteworthy component of phony users.

4.1.3. Number of hashtags

Users can exhibit trending topic with hashtag (#) symbol to a tweet. Hashtags (#) are the most-said terms on Twitter right then and there, this week or this month. On the off chance that there are numerous tweets containing a similar term, the term will turn into a trending topic. Fake users frequently post numerous irrelevant tweets that contain the trending topics to bait genuine users to peruse their tweets. Twitter thinks about an account as fake user "if a user presents different inconsequential updates on a subject utilizing the #hashtag". The quantity of tweets which is utilized as one of the substance-based highlights in. In any case, number of hashtags is used as a feature for classification in our approach.

4.1.4. Number of mentions

A unique username identified and referred with @username format in tweets on Twitter. Every user can send a reply to anyone whether these users are his or her friends or not with @username+message format where @username is the message collector. All tweets containing a username in the @username format are automatically gathered by Twitter. Every user not only can track discussion but also find each other on Twitter using the reply and mention features. In any case, spammers frequently misused this component by including numerous @username as spontaneous replies or mentions in their tweets. On the off chance that an account incorporates such a large number of replies or mentions in his tweets, Twitter will think about that account as suspicious. Therefore, number of mentions is used as a feature for classification in our approach.

4.1.5. Number of retweets

Users enable to retweet tweets that are posted from different users. User can retweet tweets with the symbol @RT. The number of retweets is utilized as one of the features in our fake account detection.

4.1.6. Number of fake words

Malicious users used fake words in their tweets. Former researchers find these fake words from spam words list, but spam words list is based on spam email. Therefore, using these spam words is not effective for fake account detection and their detection accuracy is not good. In our approach, blacklist is used to find fake words instead of spam words list. The creation of blacklist is discussed in Section 3. This feature is very effective in our approach.

4.1.7. Fake words ratio

Most of the fake users posts tweets with fake words. Therefore, fake words ratio is the most significance feature for fake account detection system. Fake words ratio is the ratio of the number of fake words by the total number of words in tweets. In the event that the fake words is high, the likelihood of fake is additionally high. Fake words ratio can be computed by the following Equation (1).

$$\text{FakeWordsRatio} = \frac{\text{NumberOfFakeWords}}{\text{TotalNumberOfWords}} \tag{1}$$

where,
FakeWordsRatio = fake words ratio.
NumberOfFakeWords = the number of fake words in tweets.
TotalNumberOfWords = the total number of words that the user post.

4.1.8. URLs ratio

Typeset sub-subheadings in medium face italic and capitalize the first letter of the first word only. URLs ratio can be calculated by the following Equation (2).

$$\text{URLsRatio} = \frac{\text{NumberOfURLs}}{\text{TotalNumberOfTweets}} \tag{2}$$

where,
NumberOfURLs = the number of URLs containing tweets posted by the user.
TotalNumberOfTweets = the total number of tweets posted by the user.

4.1.9. Hashtags ratio

Hashtags are used by counterfeit users to catch the eye of the authentic users with the goal that hashtags proportion can be utilized for distinguishing counterfeit users. If the proportion of the hashtags is high, then the suspect of the user is also high. The following Equation (3) is the hashtags ratio equation.

$$\text{HashtagsRatio} = \frac{\text{NumberOfHashtags}}{\text{TotalNumberOfTweets}} \tag{3}$$

where,
HashtagsRatio = hashtag ratio.
NumberOfHashtags = the number of hashtags containing in user's tweets.
TotalNumberOfTweets = total number of tweets that the user post.

4.1.10. Mention ratio

The higher the mention ratio is, the greater the user suspect. The following Equation (4) can be used to compute the mention ratio feature.

$$\text{MentionRatio} = \frac{\text{NumberOfMentions}}{\text{TotalNumberOfTweets}} \tag{4}$$

where,
MentionRatio = mention ratio.
NumberOfMentions = the number of mentions containing in user's tweets.
TotalNumberOfTweets = the total number of tweets that the user post.

4.1.11. Total number of words

Malicious users posts more words for various purposes such as to promote their products, to click their spam links, to insert porn URL links, etc. Total number of words can be used for fake identification.

4.1.12. Mean time between tweets (μ)

Some fake users are not human. They are bot and they use automatic tweeting tools to keep their Twitter account dynamic, open and responsive implies that they contribute at least exertion once a day. Fake users are transcendently observed to make posts at a quicker rate when contrasted with real users. This is a vital perception and we trust this feature would enable us to catch this automation feature. Equation (5) can be utilized to compute the mean time between tweets.

$$\mu = \frac{\sum(\text{TimeStampOfTweet}(i) - \text{TimeStampOfTweet}(j))}{\text{TotalNumberOfTweets} - 1} \tag{5}$$

where,
TimestampOfTweet(i) = the timestamp of the i^{th} tweet post by the user.
TimestampOfTweet(j) = the timestamp of the j^{th} tweet post by the user.
TotalNumberOfTweets = total number of tweets posted by the user.

4.1.13. Standard deviation time between tweets (σ)

Some fake users are bot and they have automation behavior for tweeting post. Notwithstanding, ordinary users have abnormal

behavior. Therefore, this is the key feature to distinguish between fake and legitimate users. Equation (6) is the standard deviation between tweets.

$$\sigma = \sqrt{\frac{\sum(X-\mu)}{\text{TotalNumberOfTweets}-1}} \quad (6)$$

where,
σ = the standard deviation between tweets.
X = the timestamp of tweet.
μ = the mean time between tweets.
TotalNumberOfTweets = the total number of tweets posted by the user.

4.1.14. Extreme idle duration time between tweets (Idle)

Fake users are believed to be discrete in their posting tweets. Sometimes, they post tweets in blasts. This component would empowers us to get the level of progress of this lead among fake and genuine users. We can find extreme idle duration time between tweets using Equation (7).

$$\text{Idle} = \frac{\text{Max}(\text{TimeStampOfTweet}(i) - \text{TimeStampOfTweet}(j))}{\text{TotalNumberOfTweets}-1} \quad (7)$$

where,
TimeStampOfTweet(i) = the timestamp of the i^{th} tweet post by the user.
TimeStampOfTweet(j) = the timestamp of the j^{th} tweet post by the user.
TotalNumberOfTweets = the total number of tweets posted by the user.

4.2. Classification

After features extraction, DECORATE is used to recognize the fake accounts from legitimate account. DECORATE is a meta-learner for building diverse ensemble of classifier using specially constructed artificial training examples [15]. In DECORATE, ensemble is generated iteratively. It initially takes in a classifier and afterward adding it to the present ensemble. At the starting, the ensemble contains the classifier trained on the given training data. At each successive iteration, the classifiers are trained on the original data combined with artificial information. Artificial training examples are produced from the distribution of data in each iteration. R_{size} is the number of training examples. The labels for these artificial training examples are chosen in a different way from the current ensemble's predictions. Diversity data are the labeled artificially generated training data [16]. The original training data and the diversity data are combined and trained in a new classifier and achieved in a diverse ensemble. By adding this classifier to the current ensemble would increase its diversity. The more diversity, the more conserved the training accuracy. A new classifier is removed if it decreases the accuracy. This is a repetitive process and ends the process if it achieves the required committee size and reaches the maximum iteration. To classify an unlabeled example, x, the following method can be utilized. Each base classifier, C_i, in the ensemble C^* yields probabilities for the class membership of x. If $P_{C_i,y}(x)$ is the estimated probability of example x belonging to class y according to classifier C_i, then the class membership probabilities is computed for the entire ensemble as:

$$P_y(x) = \frac{\sum_{C_i \in C^*} P_{C_i}(x)}{|C^*|}$$

where,
$P_y(x)$ is the probability of x belonging to class y. The most probable class is selected as the label for x.

$$C^*(x) = \arg\max_{y \in Y} P_y(x)$$

5. EXPERIMENTAL RESULTS

For experiment, we use core i3 processor, 2 GB RAM, 500 GB HDD and 32 bit Window 7 OS. The proposed system is implemented with Java programming language (NetBeans IDE 8.2). In this paper, 1KS-10KN dataset and Social Honeypot dataset are used to test the system. 1KS-10KN dataset contains 11,000 users (1000 fake users + 10,000 normal users) and 1,354,616 tweets of these users. Social Honeypot dataset contains the information of 41,499 users and 5,613,166 their tweets. 22,223 spammers and 19,276 legitimate users are involved in this dataset. The crawling time of this dataset is 7 months from December 2009 to August 2010. Performances evaluations are based on tenfold cross validation. Decorate classifier is applied for fake accounts detection. Precision, recall and F-measure are calculated to compare the results of the approach using blacklist and the approach using spam words list. We describe the comparative results of fake account detection according to blacklist based approach and spam words list based approach testing on 1KS-10KN dataset in Figure 3. The comparison of blacklist based approach and spam words list based approach using Social Honeypot dataset is shown in Figure 4. While spam word list based approach tested on 1KS-10KN dataset achieves 0.854 for precision, 0.904 for recall and 0.8797 for F-measure; precision, recall, F-measure of blacklist

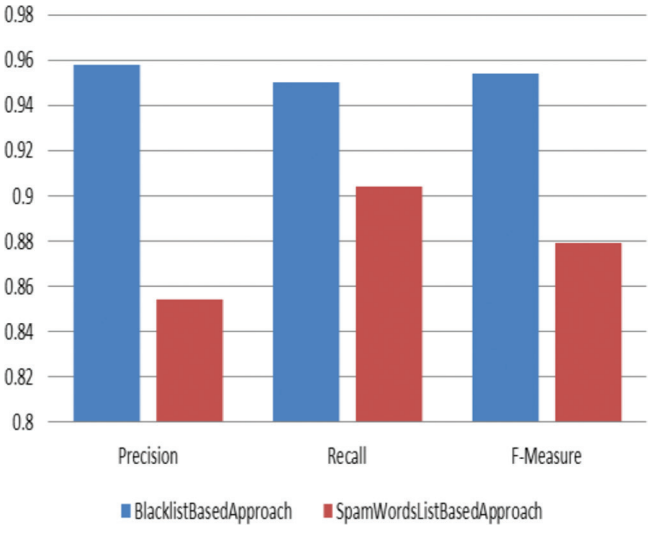


Figure 3 | Comparison of evaluation results between blacklist based approach and spam words list based approach on 1KS-10KN dataset

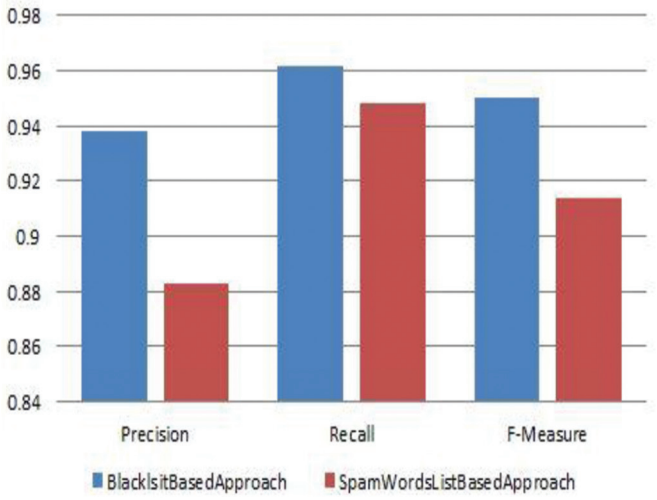


Figure 4 | Comparison of evaluation results between blacklist based approach and spam words list based approach on Social Honeypot dataset

based approach tested on 1KS-10KN dataset are 0.958, 0.95 and 0.954. When testing on Social Honeypot dataset, precision, recall and F-measure of blacklist based approach are 0.938, 0.962 and 0.950. But, precision, recall and F-measure of spam word list based approach tested on Social Honeypot dataset are 0.883, 0.948 and 0.914. Our blacklist based approach achieves higher precision, recall and F-measure than the spam words list based approach on both datasets. Therefore, the blacklist based approach is more reliable for detecting fake accounts on Twitter than the spam words list based approach. By analyzing these experimental results, it can be seen clearly that our blacklist based approach is more reliable for fake accounts detection than the traditional spam words list approach.

According to the experimental result, blacklist based approach is better accuracy than the spam words list based approach. When testing on Social Honeypot dataset, the detection rate of blacklist based approach is 95.4% that is higher than that of spam words list based approach testing on social. Using 1KS-10KN dataset for testing, detection rate of blacklist based approach is 95.4% and detection rate of spam words list based approach is 87.5%. False positive rate of spam words list based approach is 0.154 and false positive rate of blacklist based approach is 0.042. Therefore, our blacklist based approach is more effective than the approach using traditional spam words list. To reduce false positive rate is very important work in fake account detection system because the detection system incorrectly predicts real user to fake user, it can defame that user. Our approach significantly reduces the false positive rate rather than the traditional spam words list based approach. Therefore, this approach is more secure than the spam words list based approach.

6. CONCLUSION

In this paper, a new and robust blacklist creation for detecting fake accounts on Twitter is proposed. The blacklist is created using latent Dirichlet allocation and TF–IDF methods. Decorate classifier is applied to test the proposed approach. The blacklist based approach achieves acceptable accuracy and reduces false positive rate.

Table 1 | Confusion matrix for blacklist based approach on Social Honeypot dataset

		Predict	
		Fake	**Normal**
Actual	Fake	481	19
	Normal	32	468

Table 2 | Confusion matrix for spam words list based approach on Social Honeypot dataset

		Predict	
		Fake	**Normal**
Actual	Fake	474	26
	Normal	63	437

Table 3 | Confusion matrix for blacklist approach on 1KS-10KN dataset

		Predict	
		Fake	**Normal**
Actual	Fake	475	25
	Normal	21	479

Table 4 | Confusion matrix for spam words list approach on 1KS-10KN dataset

		Predict	
		Fake	**Normal**
Actual	Fake	452	48
	Normal	77	423

The proposed approach utilizes only the content of the users. The contents of the users are the text of the tweets. In this approach, profile and network data are not needed to retrieve; therefore, this can reduce time and cost overhead for extracting these features. Experimental results show that our blacklist based detection approach give better accuracy than the traditional spam words list approach. Twitter is ever changing and most of the fake users change their attacking patterns. In future, we will find more robust features that will cover the various attacking pattern of fake users.

REFERENCES

[1] Z. Chu, S. Gianvecchio, H. Wang, S. Jajodia, Detecting automation of twitter accounts: are you a human, bot, or cyborg?, IEEE Transactions on Dependable and Secure Computing, IEEE, IEEE Computer Society, 2012, pp. 811–824.

[2] A.H. Wang, Don't Follow Me: Spam detection in Twitter, Proceedings of the International Conference on Security and Cryptography (SECRYPT), IEEE, Athens, Greece, 2010, pp. 68–73.

[3] L. Liu, Y. Lu, Y. Luo, R. Zhang, L. Itti, J. Lu, Detecting "Smart" spammers on social network: a topic model approach, 2016, pp. 45–50.

[4] F. Benevenuto, G. Magno, T. Rodrigues, V. Almeida, Detecting Spammers on Twitter, Proceedings of the 7th Annual Collaboration, Electronic Messaging, Anti-abuse and Spam Conference (CEAS), Rddmond, Washington, United State, Vol. 6, 2010.

[5] M. McCord, M. Chuah, Spam detection on twitter using traditional classifiers, international conference on autonomic and trusted computing, Springer-Berlin, Heidelberg, 2011, pp. 175–186.

[6] A.A. Amleshwaram, N. Reddy, S. Yadav, G. Gu, C. Yang, Cats: characterizing automation of twitter spammers, 2013 Fifth International Conference on Communication Systems and Networks (COMSNETS), IEEE, Bangalore, India, 2013.

[7] C. Yang, C.R. Harkreader, G. Gu, Die free or live hard? Empirical evaluation and new design for fighting evolving Twitter spammers, in: R. Sommer, D. Balzarotti, G. Maier (eds.), Recent Advances in Intrusion Detection, Springer, Berlin, Heidelberg, 2011, pp. 318–337.

[8] C. Meda, F. Bisio, P. Gastaldo, R. Zunino Diten, Machine Learning Techniques Applied to Twitter Spammers Detection, 2014 48th Annual IEEE International Carnahan Conference on Security Technology, IEEE, Rome, Italy, 2014, pp. 1–6.

[9] A. Chakraborty, J. Sundi, S. Satapathy, SPAM: A Framework for Social Profile Abuse Monitoring in CSE508 Report, Stony Brook University, Stony Brook, NY, 2012.

[10] K. Lee, J. Caverlee, S. Webb, Uncovering social spammers: social honeypots + machine learning, Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, New York, NY, USA, 2010, pp. 435–442.

[11] G. Gee, H. Teh, Twitter spammer profile detection, 2010.

[12] P.C. Lin, P.M. Huang, A Study of Effective Features for Detecting Long-surviving Twitter Spam Accounts, 2013 15th International Conference on Advanced Communication Technology (ICACT), IEEE, PyeongChang, South Korea, 2013.

[13] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, J. Mach. Learn. Res. 3 (2003), 993–1022.

[14] A. Rajaraman, J. Ullman, J. Leskovec, Data Mining: Mining of Massive Dataset, Cambridge university press, 2011, pp. 1–17.

[15] P. Melville, R.J. Mooney, Constructing Diverse Classifier Ensembles Using Artificial Training Examples, Proceedings of the International Joint Conference on Artificial Intelligence, vol. 3, 2003, pp. 505–510.

[16] D. Zhang, M.M. Islam, G. Lu, A review on automatic image annotation techniques, Pattern Recognit. 45 (2012), 346–362.

[17] J. Beel, B. Gipp, S. Langer, C. Breitinger, Research-paper recommender systems: a literature survey, Int. J. Digital Libr. 17 (2015), 305–338.

[18] The list of email spam trigger words, Available from: http://blog.hubspot.com/blog/tabid/6307/bid/30684/The-Ultimate-List-of-Email-SPAM-Trigger-Words.aspx.