Automatic Detection of Cyber Security Related Accounts on Online Social Networks: Twitter as an example

Çağrı B. Aslan
Ankara Yildirim Beyazit University
Ankara, Turkey
cbaslan@ybu.edu.tr

Rahime Belen Sağlam
Ankara Yildirim Beyazit University
Ankara, Turkey
rbsaglam@ybu.edu.tr

Shujun Li
University of Kent
Canterbury, UK
www.hooklee.com

ABSTRACT

Recent studies have revealed that cyber criminals tend to exchange knowledge about cyber attacks in online social networks (OSNs). Cyber security experts are another set of information providers on OSNs who frequently share information about cyber security incidents and their personal opinions and analyses. Therefore, in order to improve our knowledge about evolving cyber attacks and the underlying human behavior for different purposes (e.g., crime investigation, understanding career development of cyber criminals and cyber security professionals, detection of impeding cyber attacks), it will be very useful to detect cyber security related accounts on OSNs automatically, and monitor their activities. This paper reports our preliminary work on automatic detection of cyber security related accounts on OSNs using Twitter as an example. Three machine learning based classification algorithms were applied and compared: decision trees, random forests, and SVM (support vector machines). Experimental results showed that both decision trees and random forests had performed well with an overall accuracy over 95%, and when random forests were used with behavioral features the accuracy had reached as high as 97.877%.

CCS CONCEPTS

• **Information systems → Social networks**; • **Security and privacy → Human and societal aspects of security and privacy**; • **Computing methodologies → Machine learning**;

ACM Reference Format:
Çağrı B. Aslan, Rahime Belen Sağlam, and Shujun Li. 2018. Automatic Detection of Cyber Security Related Accounts on Online Social Networks: Twitter as an example. In *SMSociety '18: International Conference on Social Media and Society, July 18–20, 2018, Copenhagen, Denmark.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3217804.3217919

1 INTRODUCTION

Today, online social networks (OSNs) are actively being used by a huge number of users to share opinions and information on different topics virtually over the Internet. They have the power of gathering opinions, experiences or attentions in response to real-world events from people all around the world. They can be great sources of useful information if effective algorithms are used. By acquiring such rich data sources, it has been proven possible to obtain valuable information about the effect and characterization of many different fields such as natural disasters, disease epidemics, and cyber attacks [1–3]. In this context, automatic user (account) detection (classification) has been a problem of significant interest among researchers interested in OSNs. Twitter, one of the most popular OSNs, has been subject to many such research studies, and is used as the example platform in this paper as well.

One of the studies on automatic account classification was done by Pennacchiotti et al. in 2011 [4], with the aim of classifying user accounts according to their interests on Twitter. The features used fall into four categories: profile, messaging behavior, linguistic, and social network features. The gradient boosted decision trees were used as the classification algorithm and it was reported that the system had achieved good precision and recall values above 80%. Within the study, the researchers proposed a methodology to identify prototypical words that are typical lexical expressions for people in a specific class, which are then used as linguistic features for classification. In our study, we have adapted Pennacchiotti et al.'s prototypical words algorithm to extract typical keywords used by cyber security experts who are active on Twitter.

A major research topic related to automatic detection of cyber security accounts is spam/spammer detection. Cyber criminals have been using Twitter and other OSNs to send spam, to conduct phishing attacks, and to spread malicious code [5–7]. Consequently, discovering spammer accounts that conduct such activities has been the subject of many studies. Many of these studies use profile features (e.g., number of tweets, number of followers/followees, age of the account), content features (e.g., n-grams, topic models) and linguistic features for the detection task [4, 7, 8]. In this context, the most commonly used machine learning methods include SVM (support vector machines), decision trees and random forests.

In a recent study, Aswani et al. proposed an approach to identifying spam profiles (fake followers) by combining social media analytics and bio-inspired computing [9]. They have used a set of 21 metrics under two categories (user and content based metrics), most of which are also used in this study. It was reported that the k-means integrated levy flight firefly algorithm (LFA) had produced the best result with an accuracy of 97.98%.

Similar features have been leveraged by Adewole et al. with the aim of both spam and spammer detection [10]. The researchers applied a bio-inspired evolutionary search algorithm to identify reduced features for spammer detection and ran several classification algorithms where random forests were shown to be the best.

SMSociety '18, July 18–20, 2018, Copenhagen, Denmark

ACM ISBN 978-1-4503-6334-1/18/07.
https://doi.org/10.1145/3217804.3217919

In this study, predefined spam words have also been used as content based features. In our study we have extracted cyber security related keywords automatically using three different techniques.

To the best of our knowledge, although there is plenty of work on spam and spammer classification, there is very little work on classification of cyber security related accounts. One such work was done by Lee et al. with the aim of grouping a set of experts who have highly contributed in the cyber security domain on Twitter and RSS blogs [11]. Their primary research concern was to detect emerging topics on cyber security with the help of automatically uncovering diverse experts as new information providers. Besides using a list of cyber security experts as seed accounts, they also selected a number of new accounts that were mentioned in tweets or retweets posted by accounts in that list. The accounts that have high topic relevance are determined based on the number of related topics given a predefined topics list. Those new accounts are then verified manually and added to the list of seed accounts. The whole list of cyber related accounts is used to detect emerging new topics. Since the identification of cyber security related accounts is not the main concern in Lee et al.'s study, this step is performed as a function of querying the account's articles containing the given predefined seed topics implemented in elastic search, which can limit the success of the study due to the dependencies on the comprehensiveness and representativeness of the predefined topics and the evolving nature of the cyber security domain. As a comparison, our methodology reported in this paper does not require manual extraction of any cyber security related topics or keywords.

Another set of related work is on forensic analysis of cyber criminals' activities on OSNs. For instance, Lau et al. reported a weakly supervised machine learning method for detecting criminal networks on OSNs including Twitter and online forums [12]. Their work focuses on conversational messages linking different users so the topic differs from ours reported in this paper.

2 OUR WORK

Due to the huge number of accounts on OSNs, it can take enormous human effort and normally requires some domain-specific knowledge to uncover accounts used by cyber security experts. In this paper, such a task is done automatically using a machine learning algorithm that learns a classification model from labeled data with 22 selected metrics grouped under three main categories (profile features, content features, and behavioral features). Unlike the current study, none of the previous work has taken into consideration the following two sets of features: lexical diversity, and cyber security related keywords *automatically* extracted using 3 different methodologies (prototypical words, weirdness and tf-idf values). We use Twitter as an example platform, but the features and methods reported in the paper can be easily generalized to other OSNs.

2.1 Machine Learning Model

Main text categorization methods in the literature include SVM, decision trees and random forests. In this work, these algorithms were leveraged and their performances were compared in terms of accuracy, precision, recall and F1 scores. Implementation was done using the scikit-learn tool [13]. Experimental results were based on 4-fold cross validation of the data. Different experiments were designed to cover different sets of features to evaluate the importance of the features in the classification task. The radial basis function (RBF) kernel is used for SVM. There are two parameters, which are c and γ, to be considered while using the RBF kernel with SVM. The parameter γ basically defines how each one of the training examples influence the results while the parameter c defines the smoothness of the decision surface. In this work, c was set to 1.0 and γ was set to auto mode of scikit-learn.

2.2 Dataset

We were not aware of any public dataset of cyber security related accounts on Twitter, so we developed a crawler in Python, which takes advantage of the Twitter API. Cyber security related accounts were determined via Twitter lists created by Twitter users. Twitter lists can be perceived as a way of tagging people based on their interests. Cyber security related keywords like "hackers" and "cyber security professionals" were used to identify cyber security related lists manually by the authors of the paper. Those lists were then analyzed by an independent cyber security professional (who is not a co-author of the paper and had been working as a cyber security consultant for 7 years). As a result, 212 cyber security related accounts were validated. In order to have a balanced dataset, 212 ordinary accounts that do not have a focus on cyber security were selected randomly (also manually by the authors of the paper). Timelines of all the 424 accounts were crawled by using the Twitter API, leading to 3,200 tweets per account (which is the maximum number of tweets we could extract per account through the API without paying a fee).

2.3 Features Used

For the classification task, we used 22 metrics to produce features of a given account on Twitter. Note that one metric may lead to more than one feature: three metrics (tf-idf, weirdness, prototypical words) correspond to a number of features per metrics (only one of the metrics is used as they are different ways to generate keywords as features), and the remaining $22 - 3 = 19$ metrics correspond to only one feature per metric. Table 1 lists all the metrics under three categories, which are explained in more details below. Three techniques were used for keyword extraction: prototypical words approach, weirdness scores and tf-idf scores. Based on these techniques three different keyword lists (each contains 200 words, 100 for the cyber security related class and 100 for the non-cyber security related class) were extracted from the training sets. These lists were used to generate feature sets for each user by calculating the term frequency of each keyword in the lists by taking the number of times a keyword occurs in a timeline divided by the total number of words in that timeline. In total we used $200 + 19 = 219$ features for the classification task, and the 200 keywords are generated by just one technique. We also tried merging keywords generated by all the three techniques to have an enlarged list of keywords, which gave us 504 keywords (less than $3 \times 200 = 600$ due to overlaps among the three lists) and $19 + 504 = 523$ features in total.

2.3.1 Profile Features.

Profile features were extracted from profile information that Twitter provides. In the literature it is a very common practice to use profile features and features originated

Table 1: Feature list

Profile Features	Behavioral Features	Content Features
number of alphabetic characters	number of tweets	lexical diversity
number of numeric characters	number of retweets	Flesch-Kincaid score
number of capitalization	average number of hashtags	SMOG index
number of friends	average number of urls	prototypical words
number of followers	average time between tweets	weirdness score
friends/followers ratio	standard deviation between tweets	tf-idf
use of the avatar picture	fraction of tweets posted in 24 hours	
presence of location		
length of user name		

from profile information. These features are the very first step to represent users in Twitter since they do often give distinctive information about the user. For instance, usage of the numeric characters instead of some letters is a common practice among cyber security related users. Two concrete examples are: 3 (which is basically the reverse of the letter 'E') is used instead of "e", and 4 is used instead of "a" (which look alike in many font styles).

2.3.2 Content Based Features. These features were extracted from the content of the user's Twitter feed and represent user's lexical usage and main interests. This study used a set of 6 metrics under this category including lexical diversity, Flesch-Kincaid grade level score, SMOG (Simple Measure of Gobbledygook) index, frequency of some keywords extracted using three different techniques (prototypical words, weirdness score and tf-idf).

The lexical diversity is the percentage of unique words or terms out of total words in a document. It has been used in the literature for spam detection [9]. The Flesch-Kincaid grade level and SMOG are two popular readability formulas that measure the text difficulty through sentence and word length.

Prototypical words were extracted following the method reported in [4]. Prototypical keywords can be good features for such classification tasks because they relate to typical lexical expressions for users in a specific class. Given n classes, each class c_i is represented by a set of seed users S_i. Each word w is assigned a score for each class that estimates the conditional probability of the class given the word as follows:

$$\text{proto}(w, c_i) = \frac{|w, S_i|}{\sum_{j=1}^{n} |w, S_j|}, \tag{1}$$

where $|w, S_i|$ is the number of times the word w is issued by users for class c_i. In our study, we chose the highest scoring 100 words as the prototypical words. A user u is assigned a score for each prototypical word w which is computed as follows;

$$\text{f\_proto\_wp}(w, u) = \frac{|w, w_p|}{\sum_{w \in W_u} |u, w|}, \tag{2}$$

where $|w, w_p|$ is the number of times the prototypical word w is issued by user u, and W_u is the set of all words issued by u. In order to evaluate the performance of the prototypical words in the cyber security related account classification task, we extracted keywords with two other techniques as well including weirdness and tf-idf scores.

Weirdness is a termhood-based method that relies on the assumption that the distributions of terms in a specialized corpus (domain) and in a general corpus (background) differ significantly from each other [14]. It is expressed by the following formula:

$$\text{weirdness} = \frac{f_s(i)}{n_s} \bigg/ \frac{f_g(i)}{n_g}, \tag{3}$$

where $f_s(i)$ and $f_g(i)$ are the frequencies of the i-th word in the specialized and the general corpus, respectively, n_s is the total number of words in specialized corpus and n_g is the total number of words in the general one.

The tf-idf feature is the acronym of "term frequency - inverse document frequency" widely used for information retrieval tasks. It is defined as the product of the term frequency $\text{tf}_{w,d}$, which is defined here as the raw count of word w in a document d divided by the total number of words in d, and the inverse document frequency defined by the following formula:

$$\log \frac{N}{1 + N_w}, \tag{4}$$

where N is the total number of documents, and N_w is the number of documents mentioning the word w at least once. In the training phase, we considered a document to be the set of tweets shared by all the accounts belonging to a specific class. Since there are only 2 classes (cyber security related and non-cyber security related), there are only 2 documents, i.e., $N = 2$ and $N_w \leq 2$. The idf is calculated based on the training set and then used as a constant in the testing phase. For the term frequency, in the testing phase we treat each target account's timeline as the document d.

2.3.3 Behavioral Features. Behavioral features are extracted from the timeline of the target user account and provide statistics about how the user interacts with Twitter. There are five different behavioral features used in this study. Basically a user can tweet, retweet or mention other users. In addition, they can share different types of contents like a tweet with hashtags and URLs. All of these metrics were evaluated using these features.

While the numbers of retweets and hashtags may give insights about the sociability of a user, the standard deviation of the inter-tweet time interval (in seconds) and the fraction of tweets belonging to the last 24 hours (out of all tweets) can represent the level of activeness of the user. Such information is useful for training a model for classification of cyber security related accounts since

Table 2: All features

	Accuracy	F1	Precision	Recall
SVM	0.5	0.66667	0.5	1
Decision Trees	0.96462	0.96445	0.9673	0.96226
Random Forests	0.95519	0.95425	0.97058	0.93868

Table 3: Profile features

	Accuracy	F1	Precision	Recall
SVM	0.50472	0.66878	0.50238	1
Decision Trees	0.57075	0.56295	0.57049	0.5566
Random Forests	0.65566	0.67436	0.64404	0.71226

Table 4: Behavioral features

	Accuracy	F1	Precision	Recall
SVM	0.5	0.66667	0.5	1
Decision Trees	0.9717	0.97196	0.96801	0.97642
Random Forests	0.97877	0.97875	0.98122	0.97642

those users are information providers who tend to post tweets more often than information seekers.

3 RESULTS

Supervised classification of cyber security related accounts on Twitter is the main purpose of this work. For this purpose, three machine learning techniques were applied to seven different subsets of features presented in Table 1.

In order to evaluate the importance of different sets of features on the classification task, several experiments were conducted and their results are summarized in Tables 2-5, for all 523 features and different subsets (nine profile features, seven behavioral features, and 3 + 504 = 507 content based features). Table 4 shows that behavioral features in particular perform best whereas results obtained with content features are also encouraging. Table 2 shows that combining all features makes the results slightly worse.

As described above, content based features include automatically extracted keywords using three different techniques. In order to compare their performances, three different experiments were conducted for each keyword extraction technique, where only 19 + 200 = 219 features were used. The results are shown in Tables 6-8. It is clear that all the three methods worked well, although the weirdness score based method performed slightly worse.

It is clear that random forests and decision trees performed much better than SVM. SVM performed so badly maybe because the kernel we used does not match the ideal kernel describing the boundary between the two classes. The performance differences between random forests and decision trees are minimal, although it appears that random forests are more stable across all different settings and decision trees performed slightly better when all features are considered.

Table 5: Content features

	Accuracy	F1	Precision	Recall
SVM	0.69575	0.76102	0.62759	0.96698
Decision Trees	0.93632	0.93618	0.94049	0.93396
Random Forests	0.96226	0.96261	0.95394	0.9717

Table 6: Proto and other features

	Accuracy	F1	Precision	Recall
SVM	0.5	0.66667	0.5	1
Decision Trees	0.9717	0.97124	0.98075	0.96226
Random Forests	0.97642	0.97672	0.96362	0.99057

Table 7: tf-idf and other features

	Accuracy	F1	Precision	Recall
SVM	0.5	0.66667	0.5	1
Decision Trees	0.9717	0.97142	0.98103	0.96226
Random Forests	0.97642	0.97667	0.97262	0.98113

Table 8: Weirdness and other features

	Accuracy	F1	Precision	Recall
SVM	0.5	0.66667	0.5	1
Decision Trees	0.96226	0.96212	0.96241	0.96226
Random Forests	0.95755	0.95754	0.95795	0.95755

Among the three feature categories, behavioral features performed the best, followed by the content based features. It may be surprising that behavioral features alone can support the classification task so well. This may be interpreted based on a reasonable hypothesis that cyber security experts tend to use Twitter (much) more frequently than others. We plan to verify this hypothesis in our future work.

4 LIMITATIONS

Although the results of our work are very promising, we acknowledge that there are limitations of the work, which could make it harder to generalize our results; thus, further investigations are needed for these issues.

A major issue is about the "natural" ratio between cyber security related and non-cyber security related accounts on Twitter. We expect that the ratio is far from balanced (i.e., 1:1) because it is very likely that the number of non-cyber security related accounts is significantly higher than the number of cyber security related accounts. Therefore, as long as the false positive rate and the false negative rate are not equal, which is what we have observed in our experiments, the realistic accuracy of the trained classifiers will differ from what we report above, where the realistic accuracy is defined as the probability that a uniformly randomly selected Twitter account is classified correctly. This issue is actually not simple to address because "accounts on Twitter" are not well defined; one should consider only active accounts but the word "active" is vague. In addition, Twitter API does not provide a reliable way to estimate the number of active accounts, so we can only estimate it using indirect methods. In future, we plan to investigate this issue more, by developing some reasonable sampling methods and verifying the performance of the trained classifier in the wild.

Another issue is about the keywords used. The number 200 was selected heuristically, and it is probably better to make it dynamic based on some selection criteria. Having dynamic keywords is important since the cyber security domain is evolving rapidly, so new keywords keep emerging while some get out-dated quickly. This also implies that we will end up with a different number of features dynamically, so the classifier needs retraining from time to time. As a consequence, we need to look at incremental learning to keep the classification model updated. In our future work we will also look at this direction and see what we can do.

A third limitation is about the machine learning models we used. While both random forests and decision trees produced very good results, there may be other models (including hybrid models) that can perform even better. For instance, if we can collect a large database, deep learning may allow us to train a more accurate model. In addition, the poor performance of the SVM may be due to the mismatch of the kernel and parameters we used, so there is also space to improve. Our future work will also cover investigation into this line.

Yet another issue (not a limitation per se) is how we labelled a Twitter account as cyber security related. What "cyber" means is clearly not well defined, and there are many different opinions about what it constitutes. In this work, we used a single cyber security expert for the labelling task. This may make our results biased, namely, the trained classifier actually learns about that single individual's judgment rather than that of an average (typical) cyber security expert. A solution may be to ask more cyber security experts to do the labelling and use their average opinions as the ground truth. The dynamic requirement we mentioned above implies that we need to consider a human-in-the-loop approach, e.g., via expert-based crowdsourcing.

5 CONCLUSIONS

In this paper, automatic detection of cyber security related accounts on Twitter is investigated based on three different sets of feature and three different machine learning methods. Experimental results showed very promising performance with high accuracy over 95%. The highest score (over 97%) was achieved by applying the random forests method to behavioral features.

Maintaining a list of cyber security related accounts manually requires domain-specific knowledge and takes human efforts. Our work suggests that we can automatically maintain such a list, and use it for more complicated analysis on things such as cyber security related events and human behaviors of cyber criminals and cyber security experts, via automated monitoring of accounts in the list. In our future work, we will look at how to apply the work reported in this paper for automated cyber security event detection. We also call for the wider community to explore other applications of the automatic detection method in OSN research.

REFERENCES

[1] Zahra Ashktorab, Christopher Brown, Manojit Nandi, and Aron Culotta. Tweedr: Mining Twitter to inform disaster response. In *Proceedings of the 11th International Conference on Information Systems for Crisis Response and Management Conference (ISCRAM 2014)*, 2014.

[2] Kathy Lee, Ankit Agrawal, and Alok Choudhary. Real-time disease surveillance using Twitter data: Demonstration on flu and cancer. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2013)*, pages 1474–1477. ACM, 2013.

[3] Alan Ritter, Oren Etzioni, and Sam Clark. Open domain event extraction from Twitter. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2012)*, pages 1104–1112. ACM, 2012.

[4] Marco Pennacchiotti and Ana-Maria Popescu. Democrats, republicans and Starbucks afficionados: User classification in Twitter. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2011)*, pages 430–438. ACM, 2011.

[5] Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. Detecting spammers on social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference (ACSAC 2010)*, pages 1–9. ACM, 2010.

[6] Kyumin Lee, James Caverlee, and Steve Webb. Uncovering social spammers: Social honeypots + machine learning. In *Proceedings of the 33rd International ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR 2010)*, pages 435–442. ACM, 2010.

[7] Fabricio Benevenuto, Gabriel Magno, Tiago Rodrigues, and Virgilio Almeida. Detecting spammers on Twitter. In *Proceedings of 7th Annual Collaboration, Electronic messaging, Anti- Abuse and Spam Conference (CEAS 2010)*, 2010.

[8] Bo Wang, Arkaitz Zubiaga, Maria Liakata, and Rob Procter. Making the most of tweet-inherent features for social spam detection on Twitter. *ArXiv e-prints*, 2015.

[9] Reema Aswani, Arpan Kumar Kar, and P. Vigneswara Ilavarasan. Detection of spammers in Twitter marketing: A hybrid approach using social media analytics and bio inspired computing. *Information Systems Frontiers*, 2017:1–16, 2017.

[10] Kayode Sakariyah Adewole, Nor Badrul Anuar, Amirrudin Kamsin, and Arun Kumar Sangaiah. SMSAD: a framework for spam message and spam account detection. *Multimedia Tools and Applications*, pages 1–36, 2017.

[11] Kuo-Chan Lee, Chih-Hung Hsieh, Li-Jia Wei, Ching-Hao Mao, Jyun-Han Dai, and Yu-Ting Kuang. Sec-Buzzer: cyber security emerging topic mining with open threat intelligence retrieval and timeline event annotation. *Soft Computing*, 21(11):2883–2896, 2017.

[12] Raymond Y. K. Lau, Yunqing Xia, and Yunming Ye. A probabilistic generative model for mining cybercriminal networks from online social media. *IEEE Computational Intelligence Magazine*, 9(1):31–43, 2014.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[14] Raymond Y. K. Lau, Yunqing Xia, and Yunming Ye. A probabilistic generative model for mining cybercriminal networks from online social media. *IEEE Computational Intelligence Magazine*, 9(1):31–43, 2014.