Malicious Behaviour Identification in Online Social Networks

Raad Bin Tareaf(✉), Philipp Berger, Patrick Hennig, and Christoph Meinel

Hasso Plattner Institute, University of Potsdam, Potsdam, Germany
{raad.bintareaf,philipp.berger,patrick.hennig,
christoph.meinel}@hpi.uni-potsdam.de

Abstract. This paper outlines work on the detection of anomalous behaviour in Online Social Networks (OSNs). We present various automated techniques for identifying a 'prodigious' segment within a tweet, and consider tweets which are unusual because of writing style, posting sequence, or engagement level. We evaluate the mechanism by running extensive experiments over large artificially constructed tweets corpus, crawled to include randomly interpolated and abnormal Tweets. In order to successfully identify anomalies in a tweet, we aggregate more than 21 features to characterize users' behavioural pattern. Using these features with each of our methods, we examine the effect of the total number of tweets on our ability to detect an anomaly, allowing segments of size 50 tweets 100 tweets and 200 tweets. We show indispensable improvements over a baseline in all circumstances for each method, and identify the method variant which performs persistently better than others.

Keywords: Online social networks · Twitter
Anomaly detection · Authorship authentication

1 Introduction

Online Social Networks (OSNs) present convenient platforms for users to participate, interact, and collaborate in online manner. While users' relish the openness and amenity of social media, various malicious unethical activities and actions can be performed by individuals or communities to manipulate thought process of OSN users' to fulfil their own agenda. Therefore, it is extremely critical to detect these unusual activities as accurately and quickly as possible to prevent potential attacks and disasters. Such malicious behaviour needs to be controlled and its consequences should be reduced.

Social media platforms acts as a medium for communication for getting an overview of trends and current situation in various domains and locations. For instance, [2] proposed a system which helps bloggers creates an effective articles by identifying so-called non annotated audience attributes (age, gender and personality traits) for potential blogs posts, while [5] introduced a system that is

S. Bonomi and E. Rivière (Eds.): DAIS 2018, LNCS 10853, pp. 18–25, 2018.
https://doi.org/10.1007/978-3-319-93767-0_2

competent in predicting political ideology and homophily between online individuals by analyzing their twitter profiles contents.

Among this potentiality, social media evolves into an interesting target for criminal [13]. As fake accounts can be easily recognized and reported [8], the attackers tend to hack into real existing accounts and compromise profiles content. Thus, provides criminals wide range of contacts and connections to spread their tragedy, with a potentially high success rate of penetration, because those contacts have already trusted relationship with the compromised profile. In 2016, more than 600,000 Facebook profiles are compromised every day [1] By taking over legitimate accounts, the attacker can easily exploit this trust relation to serve his own intentions.

Consequently, Attackers can disseminate their malicious messages or propagate fake information to a large users base. Nevertheless, detecting compromised accounts is much harder than detecting fake accounts. In comparison, a compromised account is genuine until it is successfully attacked. This yields two main benefits for attackers. First, they can misuse the existing trust between the profile owner and their contacts. Secondly, a compromised account confers ‘normal’ behaviour, thus, it may not be blocked or deleted promptly by the operators.

2 Related Works

Social networking platforms have become a very attractive target for hackers and intruders. For instance, one could spam users with malicious messages and consequently spread harmful messages. Gianluca Stringhini [13] analysed the activities of spammers by proposing a new method called “honey profiles”, this act as a “trap” to detect and counteract at unauthorized use of information systems.

Another framework for authorship identification of online-messages was proposed by [14]. The framework was examined with online-newsgroup messages in English and Chinese language. In addition, a tool called COMPA [8] uses a feature set of meta information about the single post such as time, language and location. However, the text itself was not analysed nor considered in their proposed model [3] analysed message segments of 500 characters. They introduced a combination of supervised learning technique with n-gram analysis to verify an author of a specific text. Since posts in social networks are usually very short and often have a limited length of characters, [12] focused on authorship attribution of micro-messages, particularly on tweets. They introduced the concept of an author’s unique “signature”, specifying the features that are unique for a certain user.

Our approach focuses on capturing malicious activities by extracting all user activities within Twitter profiles. Consequently, We developed machine learning algorithm to extract 21 unique feature to be able to detect malicious behaviours and reveal compromised accounts to their owners.

3 Implementation

3.1 Dataset Acquisition

We examined two existing datasets, the Followthehashtag[1] dataset and the myPersonality[2]. Nevertheless, none of them matched our specified preferences (100 tweet as a minimum per user). Eventually, we decided to gather our own dataset as an exemplary samples from Twitter platform.

Crawled Twitter Dataset: A new dataset [7] was gathered by crawling Twitter's REST API using the Python Tweepy library[3]. The new dataset contains the tweets of the 20 most popular twitter users (with the most followers)[4] whereby re-tweets are neglected. Consequently, the dataset contains a mix of relatively structured tweets, written in a formal and informative style, and completely unstructured tweets written in a colloquial style.

In total, the dataset contains 52,542 tweets with an average number of 2627.1 tweets per user. The time difference between the first and the last crawled tweet is 1,287 days (about 3.5 years) on average; half a year for *CNN Breaking News* account and up to 7 years for *Twitter official* account. Consequently, the dataset contains user accounts with 0.13 tweets per day (*YouTube*) up to 1.13 tweets per day, while the rest accounts have in average 0.5 tweets per day. Considering these statistics, the crawled dataset comprises of a well-suited mix of diverse posting behaviours.

3.2 Features Selection

(A) Text-Specific Features: Since writing style is a broad field to analyse, [14] suggests to break it down in four types: lexical, syntactic, structural and content-specific feature styles. These styles depend for example on the gender or the educational background of the person [6,11]. Therefore, We applied the following features to analyse the writing style:

- Lexical character features: LCR features are extracted from users posts considering the: amount of characters, amount of ASCII characters, amount of ASCII upper-case characters, amount of ASCII lower-case characters, amount of digits, amount of white spaces and amount of special characters.

- Lexical word features: LWF features are extracted from users posts considering the: amount of words, amount of short words, average words length, average sentence length of characters, average sentence length of words and the amount of unique words.

[1] http://followthehashtag.com/datasets/.

[2] http://mypersonality.org/wiki/doku.php?id=download_databases.

[3] http://www.tweepy.org/.

[4] http://twittercounter.com/pages/100.

- Syntactic Features: are extracted by considering the usage of punctuations and the frequency of all used punctuations.

- Structural Features: are extracted by analyzing the total number of lines and total number of sentences within a user posts.

(B) N-Grams: For our approach, we selected two types of n-gram [3] features which are Word n-grams and Character n-grams.

(C) Post-specific Features: On one hand, each user follows a specific pattern in posting behaviour [4]. On the other hand, there is a reactional pattern to a user's posts by his/her followers [8]. By analysing these two patterns, anomalies in posts can be recognized. For our approach we decided to analyse the amount of shares for posts, the amount of likes for posts as well as the time stamps for each of the post-specific features.

All extracted features are scaled using the Python `sklearn.preprocessing` package[5] to obtain a Gaussian distribution with zero mean and unit variance. This standardization is necessary to prevent a learning algorithm's objective function from being dominated by some single features and, thus, make the estimator unable to learn from other features correctly.

3.3 Training and Prediction

Training Algorithm. We implemented an algorithm which is based on a legitimate assumption that the oldest 100 tweets are actually posted by the user. In most scenarios, this assumption holds since the probability that an account was compromised increases with the time. The longer a faked post exists, the more likely users will detect it manually, either because of the reactions of friends or by themselves while checking their profile. Moreover, an account with a little amount of tweets (100 or less) usually has a few audience. Therefore, such an account is less interesting to get compromised by attackers and the probability that one of these first tweets is a compromised is relatively low. An analysis of different initial training set sizes is presented in the evaluation section.

The proposed solution starts by increasing the number of training samples incrementally utilizing interactive machine learning approach. In the beginning, the classifier is trained with the initial and oldest 100 posts of a user timeline as positive samples. The generated model is then used to predict classes for the remaining posts which are sorted by publishing time starting with the oldest one. The algorithm breaks down the predictions into cohesive batches of posts with the same predicted class. If the first batch is predicted as legitimate posts, the existing model is trained with these posts as positive samples. If the batch is predicted as malicious, posts will be added to a list of suspicious posts and the model is trained with the next batch as positive samples. In both cases, all samples in the list of suspicious posts are classified again with the updated model. If the class of a post has changed, then it will be removed from the negative list. Consequently,

[5] http://scikit-learn.org/stable/modules/preprocessing.html.

the next iteration starts with classifying the remaining posts whose batches are not examined before. This process is repeated until there are no remaining posts. The algorithm then returns the full list of the suspicious tweets.

Figure 1 demonstrates the training algorithm for an exemplary user's timeline. In each iteration, a batch of tweets is added to the positive training samples and both classifier and predictions are updated for the remaining tweets. As it demonstrated in the figure, the new batch of tweets is framed by a solid line while the dotted box contains all positive training samples. Each column shows the state at the beginning of the corresponding iteration after selecting the new batch of training samples which depends on the updated predictions of the previous iteration.

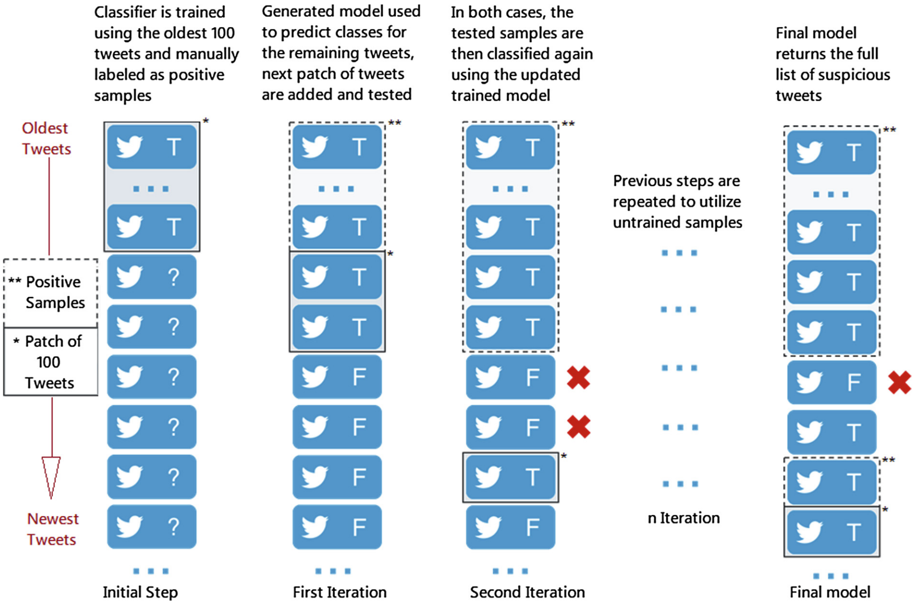
Classifier is trained using the oldest 100 tweets and manually labeled as positive samples
Generated model used to predict classes for the remaining tweets, next patch of tweets are added and tested
In both cases, the tested samples are then classified again using the updated trained model
Final model returns the full list of suspicious tweets
Oldest Tweets
Newest Tweets
** Positive Samples
* Patch of 100 Tweets
Previous steps are repeated to utilize untrained samples
n Iteration
T
?
F
Initial Step
First Iteration
Second Iteration
Final model

Fig. 1. First three iterations of the training algorithm conducted on exemplary tweets. `T` and `F` refer to the True or False predicted classes.

Model Refinement. Given the detected suspicious posts, the user who runs the algorithm can select those ones which are actually written by him/her. The marked tweets are used as labelled samples for an additional training process turning the actual classification errors into a chance to refine the generated model on fly. The existing classifier is fitted with the samples using a higher weight than in the initial training. The list of suspicious posts is then updated using the improved model.

Classifier Selection. The goal of this implementation is to procreate an architecture which does not depend heavily on a specific learning algorithm. This makes it easy to switch classifier type and compare the performance of different algorithms. In this work, we considered four classification algorithms whose evaluation results are compared in Sect. 4.2.

4 Evaluation

4.1 Evaluation Method

In online social networks domain, there is no superior evaluation criteria available to follow within the task of malicious behaviour identification. Therefore, we followed the concept of evaluation that is proposed by [9,10]. The crawled tweets are grouped by the author and sorted by publishing time starting with the oldest one. Afterwards, the data is partitioned into a training set and a validation set as described below. Since our training approach is iterative, the boundary between trained and tested status are updated in each iteration.

Training Set: Since two out of the four classifiers: Perceptron, Decision Tree, One-Class SVM and Isolation Forest are binary classification techniques, the initial training set consists of two separate sets: positive and negative samples. Being one-class classification techniques, the SVM and the IsolationForest classifier take only the positive samples as input and neglect the negative examples.

The first 100 tweets of a chosen author are used as positive sample for the training set. Additionally, a randomly sampled subset (500 tweets) of the tweets of ten other users is used as negative sample. To evaluate the performance of our developed training algorithm, the 400 next tweets of the chosen author are added.

Validation Set: As introduced above, the 400 next tweets of the chosen user are added and they are acting as part of the training set but likewise are used for testing in the sense of determining if our developed training algorithm correctly recognizes these tweets during the incremental training. Moreover, a randomly sampled subset of the tweets for the remaining nine other users (disjoint with tweets used for the training set) is inserted into the validation set whereby the temporal order of all tweets is maintained.

4.2 Classifier Performance

In order to evaluate the performance of the proposed model, we used the accuracy, precision, recall metrics and we plotted the final f-score harmony measurement for each classifier on each feature category, as shown in Table 1.

When a bigger initial training set is given as shown in Fig. 2, the recall improves significantly whereas the precision slightly decreases. Considering that the precision is more important for our stated problem and that the possibility of containing already compromised tweets raises when using a bigger initial training set size (which are then considered to be a user's posts by the algorithm), a size above 100 tweets is not well-suited as initial training set size.

Table 1. Precision and f-score values for various classifiers options over different feature subsets.

Classifier	Precision						F-measure
	(1)[a]	(2)[b]	(3)[c]	(1&2)[d]	(1&3)[e]	All features	
Perceptron	0.78	0.82	0.24	0.81	0.77	0.83	0.70
Decision Tree	0.65	0.79	0.56	0.75	0.65	0.73	0.76
One-Class SVM	0.74	0.72	0.50	0.80	0.73	0.85	0.61
Isolation Forest	0.54	0.51	0.52	0.53	0.54	0.53	0.72

[a] 1: Text-specific, [b] 2: N-grams, [c] 3: Post-specific, [d] (1&2): combination of Text-specific and N-grams, [e] (1&3): combination of Text-specific and Post-specific

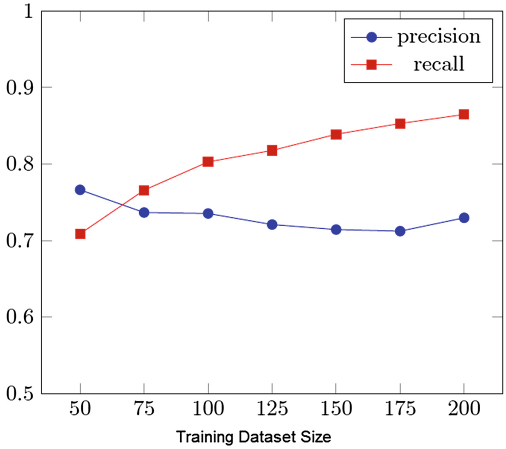

Fig. 2. Comparison of the precision and recall for different initial training set size using Decision Tree classifier.

5 Conclusion and Future Work

We presented our work of automating the process of identifying malicious behaviour in online social networks. Specifically, Twitter social platform. We extracted 21 unique features from user profiles and trained our model accordingly to characterize users' behavioural pattern and specify compromised accounts. Given a Twitter account, the proposed system can detect suspicious posts based on anomalies in user's profile and state whether the account was compromised before or not. Our novel combined features (text-specific features, n-grams feature and post-specific features) confirms that utilizing the power of machine learning classifiers can accurately detect deviations in user's posts and alert when profile behaviour is violated.

With our proposed approach, we improved the performance for specific classifiers and feature subsets by 9% (One-Class SVM) to 13% (Perceptron) while slightly lost some precision. The strength of our feature set combination is that post-specific (meta) features are considered in the experiments. In future work, our results could be improved significantly if more post-specific features, such as the geolocation are available in the dataset. The Supplementary material associated with this research is publicly available for interested researchers.

Acknowledgement. We would also like to show our gratitude to our master students (Henriette Dinger, Dominic Sauer, Soeren Oldag and Sebastian Kliem - Hasso Plattner Institute) who provided insight and expertise that greatly assisted the research during our research seminar.

References

1. Andra, Z.: 10 alarming cyber security facts that threaten your data. Heimdalsecurity (2015)
2. Bin Tareaf, R., Berger, P., Hennig, P., Meinel, C.: Identifying audience attributes: predicting age, gender and personality for enhanced article writing. In: International Conference on Cloud and Big Data Computing, pp. 79–88. ACM (2017)
3. Brocardo, M.L., Traore, I., Saad, S., Woungang, I.: Authorship verification for short messages using stylometry. In: 2013 International Conference on Computer, Information and Telecommunication Systems (CITS), pp. 1–6. IEEE (2013)
4. Corney, M., De Vel, O., Anderson, A., Mohay, G.: Gender-preferential text mining of e-mail discourse. In: 2002 Proceedings of the 18th Annual Computer Security Applications Conference, pp. 282–289. IEEE (2002)
5. Boutyline, A., Willer, R.: The social structure of political echo chambers: variation in ideological homophily in online networks. J. Polit. Psychol. **38**, 551–569 (2017). Wiley Online Library
6. De Vel, O., Anderson, A., Corney, M., Mohay, G.: Mining e-mail content for author identification forensics. ACM SIGMOD Rec. **30**(4), 55–64 (2001)
7. Bin Tareaf, R.: Tweets dataset - top 20 most followed users in Twitter social platform. In: Harvard Dataverse, V2 (2017). https://doi.org/10.7910/DVN/JBXKFD
8. Egele, M., Stringhini, G., Kruegel, C., Vigna, G.: COMPA: detecting compromised accounts on social networks. In: NDSS (2013)
9. Guthrie, D., Guthrie, L., Allison, B., Wilks, Y.: Unsupervised anomaly detection. In: IJCAI, pp. 1624–1628 (2007)
10. Guthrie, D., Guthrie, L., Wilks, Y.: An unsupervised approach for the detection of outliers in corpora. LREC (2008)
11. Koppel, M., Argamon, S., Shimoni, A.R.: Automatically categorizing written texts by author gender. Literary Linguist. Comput. **17**(4), 401–412 (2002)
12. Schwartz, R., Tsur, O., Rappoport, A., Koppel, M.: Authorship attribution of micro-messages. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 1880–1891 (2013)
13. Stringhini, G., Kruegel, C., Vigna, G.: Detecting spammers on social networks. In: Proceedings of the 26th Annual Computer Security Applications Conference, pp. 1–9. ACM (2010)
14. Zheng, R., Li, J., Chen, H., Huang, Z.: A framework for authorship identification of online messages: writing-style features and classification techniques. J. Assoc. Inf. Sci. Technol. **57**(3), 378–393 (2006)