CrowdGuard: Characterization and Early Detection of Collective Content Polluters in Online Social Networks

Ke Li
Northwestern Polytechnical University
Xi'an, Shaanxi, China,
sfk100200@qq.com

Bin Guo
Northwestern Polytechnical University
Xi'an, Shaanxi, China,
guob@nwpu.edu.cn

Qiuyun Zhang
Northwestern Polytechnical University
Xi'an, Shaanxi, China,
1975642935@qq.com

Jianping Yuan
Northwestern Polytechnical University
Xi'an, Shaanxi, China,
jyuan@nwpu.edu.cn

Zhiwen Yu
Northwestern Polytechnical University
Xi'an, Shaanxi, China,
zhiwenyu@nwpu.edu.cn

ABSTRACT

Recently, content polluters post malicious information in Online Social Networks (OSNs), which is a more and more serious problem that poses a serious threat to the privacy information, account security, user experience, etc. They continuously simulate the behaviors of legitimate accounts in various ways, and evade detection systems against them. In this paper, we focus on one kind of content polluter, namely collective content polluter (hereinafter referred to as CCP). Existing works either focus on individual polluters or require long periods of data records for detection, making their detection methods less robust and lagging behind. It is thus necessary to analyze the characteristics of collective content polluters and study the methods for early detection. This paper proposes a CCP early detection method called CrowdGuard. It analyzes the crowd behaviors of collective content polluters and legitimate accounts, extracts distinctive features, and leverages the Gaussian Mixture Model (GMM) method to cluster the two groups of accounts (legitimate users and polluters) to achieve early detection. Using the public dataset including thousands of collective content polluters on Twitter about a political election, we design an experimental scenario simulating early detection and evaluate the performance of CrowdGuard. The results show that CrowdGuard outperforms existing methods and is adequate for early detection.

CCS CONCEPTS

• Information systems ~ Clustering • Information systems ~ Social networks • Security and privacy ~ Social network security and privacy • Computing methodologies ~ Mixture modeling

WWW '19, May 13, 2019, San Francisco, USA

ACM ISBN 978-1-4503-6675-5/19/05.
https://doi.org/10.1145/3308560.3316452

KEYWORDS

Social Media, Collective Content Polluters, Early Detection, Crowd Computing, Gaussian Mixture Model

ACM Reference format:

Ke Li, Bin Guo, Qiuyun Zhang, Jianping Yuan and Zhiwen Yu. 2019. CrowdGuard: Characterization and Early Detection of Collective Content Polluters in Online Social Networks. In *Proceedings of WWW '19: The Web Conference (WWW '19), May 13, 2019, San Francisco, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3308560.3316452

1 Introduction

While online social networks (OSNs), such as Twitter, Facebook and Linked-In, offer a variety of conveniences and meet people's needs, it has become a profitable platform for attackers because of its huge user base [1]. By creating a large number of fake accounts or hijacking legitimate accounts, attackers post malicious information such as advertisements, pornography and phishing [2]. These fake accounts [3] and compromised accounts [4, 5], especially those that publish malicious information, are called *content polluters*. These malicious behaviors pose a serious threat to legitimate users' privacy information, account security and user experience [6]. Thus, it is necessary to characterize these content polluters in detail and propose automatic detection methods.

Content polluters have different forms, such as social bots [7], Sybil accounts [8], spammers [9], compromised accounts [4, 10], spam campaign [11], and so on. Among them, social bots which are controlled by software are widely adopted by malicious users, since they can be automatically created and generate and disseminate many spam messages easily. With so many forms, these content polluters have different behavior patterns and features, which poses a challenge to research techniques for detecting them.

Facing a variety of detection technology, content polluters must simulate the behaviors of legitimate accounts, so as to

evade detection. Moreover, they always evolve as detection techniques evolve. Recently, a new wave of content polluters is rising [12], called *collective content polluters*. Similar to collective anomalies in anomaly detection field [13, 14], a single CCP mixed in a legit population seems normal, but the occurrence of CCPs together as a collection is anomalous. They perform malicious actions for the same purpose, show similar group behaviors, and are usually event-driven, such as triggered by a malicious mission.

Existing works [15, 16, 17] pay more attention to individual polluter rather than CCP. They generally use the existing data to extract individual features, train a classifier, and then input unknown accounts one by one for detection. However, for CCP, most one-by-one detection methods do not work well, because they cannot capture the group dynamics. Therefore, Cresci et al. [18, 19] study the collective content polluters in a group fashion but their detection method requires long periods of data to distinguish them. Usually, the polluters are detected after the malicious event, and the legitimate users still suffer.

Similar to how law enforcement officers mark and monitor suspects and effectively give early warning before their crimes become apparent [20, 21], in this paper, we propose an *early detection* scenario and a CCP early detection method called *CrowdGuard*. At first, we analyze the group behavior of CCPs and legitimate accounts from multiple perspectives, and get some meaningful findings, which is helpful for early detection of CCP. Then, we extract both individual and group features from different aspects, including user demographics, social network statistics, interaction, content and temporal features. Next, we leverage the Gaussian Mixture Model (GMM) to cluster the two groups (i.e. CCP group and legitimate group, the same below) of accounts to achieve early detection. Finally, using the public dataset about a political election consisting of 991 CCPs and 1083 legitimate accounts on Twitter, we design an experimental scenario simulating early detection and evaluate the performance of our method, CrowdGuard. It achieves more than 98% precision and 94% recall, which indicates that it is adequate for early detection.

We summarize the major contributions of this paper as follows:

(1) We make an in-depth analysis of the differences between collective content polluters and legitimate accounts, and obtain some interesting findings.
(2) We extract 43 individual/group features, and choose GMM to describe these accounts and propose a group detection scheme.
(3) An experimental scenario simulating early detection over a public dataset is designed to evaluate the performance of our method. The results indicate that it is adequate for early detection.

2 Preliminary

We utilize public datasets from [22], part of which are used in our research. The authors focus on some social bots on Twitter during the Mayoral election in Rome in 2014. One of the runners-up hired about a thousand social bots to advertise his policies and help his campaign. After data processing, our existing dataset consists of 991 CCPs and 1083 legitimate accounts. The posting behavior of one CCP is very similar to that of a legitimate account, and they only post a few tweets a day, mainly retweet or copy the tweets of some celebrities. Their profiles are detailed, and it's hard to tell the difference only from their profiles. But the exception is that every time the candidate posts a tweet, they will retweet or copy it within a short period of time, which is just the characteristic of collective content polluters. So, any account that retweets or copies the candidate's tweets will be treated as a suspicious account and its information will be collected. Then 991 social bots will be screened out by manual evaluation, which acts as CCP in our research.

Each account contains user's basic information (friends count, followers count, create time and so on) and its tweets history (each of tweets contains content, URL, mention, hashtag, time stamp and so on) containing up to 3250 recent tweets. That is, one user corresponds to multiple tweets.

Social Fingerprinting [19] is our strong baseline. The authors introduced a bionic technique to model users' online behaviors by so-called "digital DNA" sequence. Each tweet is encoded as a nucleotide base based on its type (simple tweet, reply and retweet), and each account is encoded as a DNA strand (i.e. a string). Intuitively, a group of users who share a longer substring has higher abnormal similarity. To measure their similarity, the authors leverage a generalization of Longest Common Substring (LCS) which can be calculated by a parallel algorithm in [23], and the latent longest DNA substring shared by k accounts in a group is calculated (for each k: $2 \le k \le M$, where M is the group size). Thus, each k value corresponds an *LCS* value, and LCS curve can be plotted (see Figure 1(a)). For legit group, the LCS curve presents an exponential decay trend with the increase of k, and the *LCS* value has dropped to a very low level when k value is small. However, for CCP group, due to the similar behavior pattern, the *LCS* keeps a higher level than that of legit group, and the LCS does not drop sharply until k approaches M. In this way, the LCS curves of the two groups are different. The greater the difference, the easier it is to distinguish the two groups of accounts. Based on this, the detection method is raised. Finally, it can achieve the precision and recall of around 97%.

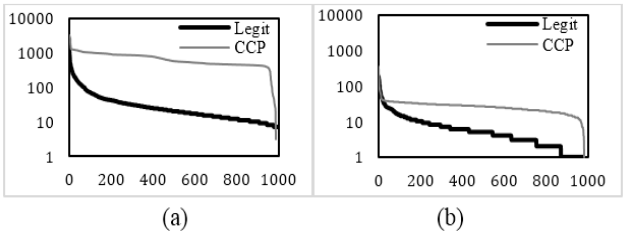

Figure 1: LCS curve in different scenario, where the Y-axis is in log scale. (a) post-event scenario. (b) early scenario.

However, defects still exist. (1) The LCS itself needs to match strings exactly, and the content polluters are judged too strictly.

(2) It is only applicable to long strings (post-event scenario). Figure 1(b) shows the LCS curve in early detection scenario, where most of the DNA chains are short, and the LCS curves of the two groups are less different than those of original scenario. Therefore, abnormal feature fragments are more difficult to be captured, and only a small proportion of content polluters can be detected, resulting in low recall, which will be shown in Section 5. (3) Although group detection is a good idea, the model is too simple and takes too few factors into consideration. Only tweet type or entity information is considered, while content, time and other information are not involved in the model. When applied to the early detection scenario, Social Fingerprinting performs poorly, that is, it is not robust enough.

To address the above defects, we will analyze CCPs in depth, and propose a targeted detection method based on this.

3 Data Analysis and Feature Extraction

3.1 Analysis of Daily Activeness

At first, we define the *daily activeness* of an account m in a group G ($1 \le m \le M$, where M is the group size, and G can be CCP or legit). It is the frequency of all the online activities (simple tweets, replies and retweets) of that account on a given day t, referred to as $A_m(t)$. Then the *daily activeness per capita* is defined as the following formula.

$$APC(t) = [\sum_{m=1}^{M} A_m(t)] / [\sum_{m=1}^{M} I(A_m(t) > 0)]$$

where $I(x)$ is an indicator function of a logical expression x. In fact, $APC(t)$ measures daily activeness of a group. It is worth noting that the $APC(t)$ considers all active users (at least posting one tweet on the day t) rather than all users in G. Figure 2(a) and 2(b) show the daily activeness per capita of CCP group and Legit group between Aug 1, 2014 and Nov 30, 2014, respectively. It can be seen from Figure 2(a) that the daily activeness per capita remains below 10 on the whole, but within 4 days around October 30, it reaches a peak of 106. Therefore, it is speculated that the election event is likely to occur before and after October 30. That is to say, CCPs are less active when there is no event, and their daily activeness can explode in a short time when there is an event. As seen in Figure 2(b), the daily activeness per capita of legitimate accounts fluctuates around 8, with an occasional peak of 9.7. Compared with CCPs, this peak is not exaggerated. It shows that the daily activeness of normal people is random and may be disturbed by some hot events, but the daily activeness of the group tends to be stable over a long period of time.

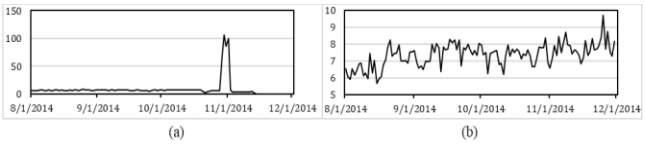

Figure 2: Daily activeness per capita between Aug 1, 2014 and Nov 30, 2014. (a) CCPs. (b) legitimate accounts.

Figure 3(a) and 3(b) give examples of four CCPs' and four legitimate accounts' daily activeness between Aug 1, 2014 and Nov 30, 2014, respectively. From Figure 3(a) we can see that at about day 90 (i.e. 2014/10/30), all of the four CCPs become active suddenly for 4 days, and they are very synchronized, which is just the collective feature. From Figure 3(b) we can see that the daily activeness curve of 4 legitimate accounts is entirely different. As an old saying goes, "No two leaves are identical in the world", which is what legitimate users should look like.

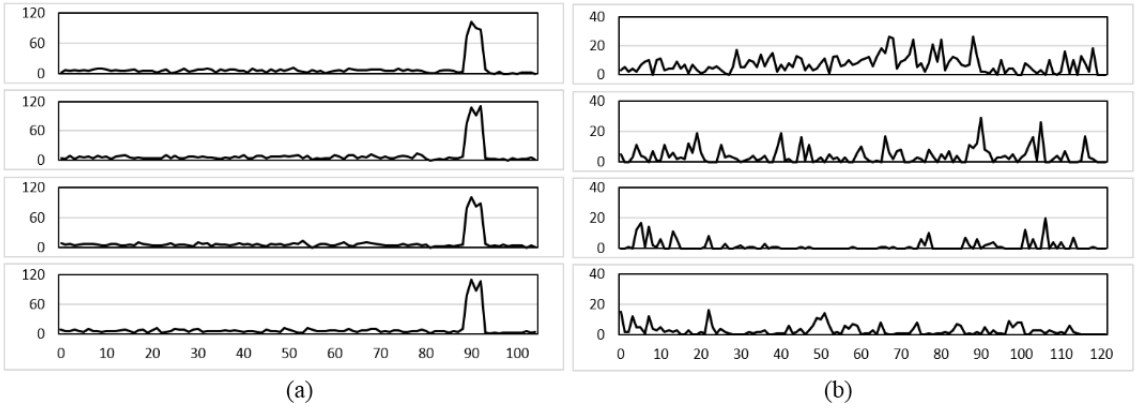

Figure 3: Examples of daily activeness between Aug 1, 2014 and Nov 30, 2014. (a) CCPs. (b) legitimate accounts.

If our application scenario is post-event, then the activeness will be a good indicator for detection. However, our application scenario is early scenario, which should be finished on the day or before the event. So, we also need to analyze the data from other aspects to find out a better indicator.

3.2 Analysis of Online Behaviors

3.2.1 Group Interaction Analysis. It mainly includes statistical analysis of replies and retweets. At first, we define the *reply ratio* of an account m as the percentage of all replies on the account to

its all tweets (considering all tweets collected, the same below). Similarly, the *retweet ratio* of an account m is the percentage of all retweets on the account to its all tweets. Figure 4(a) and 4(b) show the CDF of two groups' reply ratio and retweet ratio, respectively.

It can be seen from Figure 5(a) that most of the CCPs' reply ratio is lower than 1%, even about 40% is 0, while legitimate accounts' reply ratio is significantly higher than CCPs', and normally distributed with an average of about 0.3. This reveals that CCPs rarely interact with others. As seen in Figure 4(b), 80% of the CCPs' retweet ratio is lower than 1%, showing that they rarely facilitate the dissemination of information, especially harmless information.

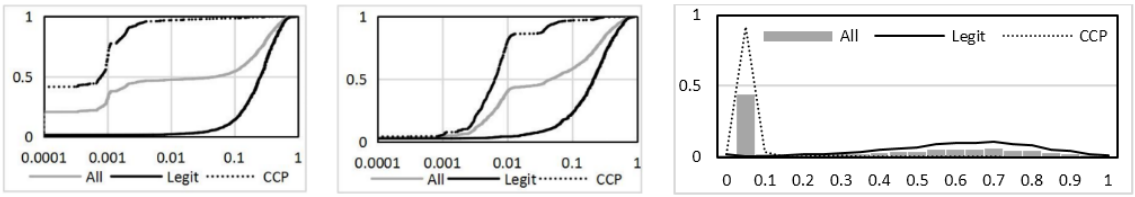

Figure 4: Distributions of reply, retweet and mention ratio, respectively. (a) CDF of reply ratio. (b) CDF of retweet ratio. (c) Histogram of mention ratio.

The above conclusions indicate that the majority of CCPs' tweets are neither replies nor retweets. Instead, they are simple tweets. The reply ratio and retweet ratio of legit group obey the normal distribution, while that of CCP group does not, and the two groups are significantly different. Moreover, legit group is loosely distributed while CCP group is more compact. So, it can be a helpful indicator for CCP detection.

3.2.2 Tweet Content Analysis. It mainly includes measures of account mention and URL usage. Similar to reply ratio, the mention ratio of an account m is defined as the percentage of all tweets with mention to its all tweets. The histogram of two groups' mention ratio is shown in Figure 4(c). From this we can see that most of the CCPs' mention ratio is lower than 0.1, while legitimate accounts' mention ratio is significantly higher than CCPs', and normally distributed with an average of about 0.7. It is also a good indicator for CCP detection.

Table 1: Online history of an example account (CCP) between April 15, 2014 and April 21, 2014.

Content	createTime
È bene fare attenzione quando tutti vi loderanno Luca evangelista	2014/4/15 10:39
http://t.co/KQY5LfaZBG	2014/4/15 14:29
Ti accorgi di toccare il fondo, quando arrivi a casa stanco e affamato, apri una bottiglia e sa di tappo!	2014/4/16 23:28
http://t.co/RZThhui7NR	2014/4/17 8:18
Hanno più foto i cuori nel caffè che io alla mia cresima.	2014/4/17 11:00
http://t.co/riNpUXaTXB	2014/4/18 7:19
Se sbaglio chiedo scusa, se ho ragione chiedo scusa comunque visto che mi porterai a sbagliare.	2014/4/18 23:38
Gli uomini sono come le salsicce:pelle fuori e maiale dentro. #sapevatelo	2014/4/19 10:49
Inutile che fate le cose di nascosto, tanto lo so che vi state organizzando, per farmi il regalo di compleanno	2014/4/19 16:10
-Voglio mostrare al mondo quello che provo per te!- ----: -Ma cosa vuoi che gliene importi al mondo!-	2014/4/19 17:09
La sensibilità è un'arma a doppio taglio : la prima lama ti apre la mente, la seconda ti squarcia il petto.	2014/4/19 22:50
Penso al senso di delusione del fondo del barile quando c'ha visto arrivare	2014/4/20 11:38
Non ogni nube porta tempesta. William Shakespeare	2014/4/20 18:48
http://t.co/G0H18Nqz2J	2014/4/21 14:29

As for URL, we observed the online history of an example account (CCP) between April 15, 2014 and April 21, 2014, in which the record is shown in Table 1. During the week, it posts 14 tweets, none of which is reply or retweet, and contains no mentions. There are 4 tweets, each of which contains only one URL, and they are basically the same format and intermingled with a few simple tweets. Considering that each of these tweets with URLs contains only one URL processed by a short URL and no other text, we can see something abnormal (legitimate users often share a link with a caption). So, we can analyze the length of all tweets with URLs (hereinafter referred to as *UTL*). Figure 5(a) shows the histogram of the two groups' UTL. Of course, a tweet contains a maximum of 140 characters. From this we can see that the UTL of legitimate accounts is evenly distributed between 20 and 130, but a large part of the population is close to 140. It is likely that one tweet is not enough to express their meaning, indicating that legitimate accounts have the intention to share as much information as possible. In contrast, nearly half of the CCPs have an UTL of 22 which is just the length of a short URL. As it turns out, CCPs send tweets with URLs without

saying another word. Thus, the accounts of the two groups are clearly differentiated in UTL.

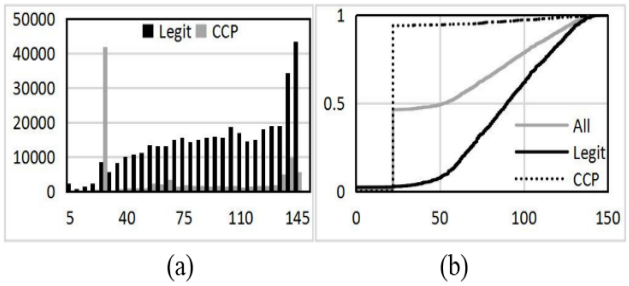

Figure 5: Analysis of URL. (a) group-wise histogram of length of tweets with URL (UTL). (b) user-wise CDF of UTL median.

Since UTL is a feature for tweets and is a distributed feature for accounts, some statistical features such as mean, median, standard deviation and so on need to be extracted. Figure 5(b) shows CDF of two groups' UTL median. More than 90% of CCPs have UTL median of 22, while the legitimate group basically presents a uniform distribution between 50 and 140. Therefore, UTL can be a good indicator for CCP detection.

3.3 Analysis of Temporal Features

Looking at Table 1, the time interval between two consecutive tweets posted by a CCP is not very short, at least three hours. Actually, legitimate accounts tend to have an online activity every few minutes [24]. So, we compute all the time intervals between two consecutive tweets of two groups, and plot the histogram in Figure 6(a), where the X-axis is in log scale of base 10 in seconds. Then, we find that both of the two groups have a bimodal distribution, but the location and size of the peaks are different. The first peak of the legit group, the main peak, is around $10^2 = 100$ seconds (several minutes), which indicates an active state. In general, a legitimate account takes about several minutes to edit and post a tweet, reply, or retweet. The second peak, also a small peak, is about $10^{4.6} \approx 40{,}000$ seconds (11 hours) which may indicate rest time. By observing the data, we find that legitimate accounts do have a segmented and continuous active state, with online behavior occurring every few minutes within each period, and the time differences between two adjacent segments are about 11 hours. For CCP group, the second peak mainly coincides with legitimate group, but the first peak is different. Its first peak is around $10^{3.9} \approx 8000$ seconds (2 hours), which is very close to its second peak, and the boundary is blurred. Therefore, CCPs don't have the so-called active state, because they are probably controlled by software. So, the time intervals of two groups are differentiated.

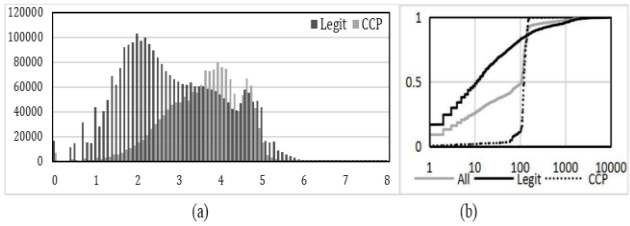

Figure 6 Analysis of temporal features, where the X-axis is in log scale. (a) group-wise histogram of time intervals between two consecutive tweets (in seconds); (b) user-wise CDF of time interval median (in minutes).

Since the time interval is a distributed feature for accounts, we will also calculate its statistical features. Figure 6(b) shows CDF of two groups' time interval median. From this we can see that most CCPs' time interval median is clustered around 100 minutes (2 hours), while legitimate group obeys the long tail distribution and is looser. So, we will select the time interval as one of the detection features.

4 The CrowdGuard Detection Method

4.1 Scenario Description and Feature Selection

As shown in section 2, our baseline in [19] only applies when historical data is sufficient, not when it is limited. It performs a detection using data up to Nov 14, two weeks after the election. By then CCPs have an adverse effect in social media environment [25]. To mitigate the negative impact, early detection is required, where CCPs can be detected at any given moment, especially on the day of the event, or even before the event. It can give early warning of catastrophic malicious events, and maintain OSN public security [26]. For example, if we want to detect at the week before the event (Oct 24), we assume that all tweets recorded by the dataset after that date are not available, only data before that date. As a result, there is no peak at the CCP group's daily activeness curve, and the amount of data available will be greatly reduced.

For an account, the number of followers, friends, age and other information can be displayed on its profile, called *explicit features*. As is shown in Section 3, reply and retweet ratio, Mention, URL, hashtag usage, time interval and so on need indirect calculation, and change dynamically over time. These features, especially those that need to be calculated from the history of tweets, are called *implicit features*. The overall scheme of our detection method, *CrowdGuard*, is that considering both of the explicit and implicit features of accounts, the mixed dataset is divided into two groups by unsupervised algorithm.

Based on the analysis in Section 3, we extract the features of account (see Table 2), and 43 features are extracted. Then we screen out the features with high degree of discrimination based on the chi-square value and the information gain [27]. As a result, the top 14 features are selected.

Table 2: List of the Features.

Type	Category	Examples	Note
Explicit	Demographics	Id, age, screen name length, description length	Age is in days, since the account created.
	Network Statistics	Followers, friends, favorites, status	Count their number and rate (subject to age)
	Boolean info	Default profile, geo enabled	
Implicit	Interaction	Activeness per day, Reply ratio, Retweet ratio	Customize the time range
	Content	URL, mention, hashtag, URL Tweet Length	Distributed features, calculating statistical features
	Temporal	Time interval between two consecutive tweets	Distributed features

4.2 Gaussian Mixture Model for Detection

As is seen in Figure 4(c), when the two groups are mixed together, they have a bimodal distribution, and each of the peaks represents a group. Not only mention ratio, but also most of the other features are similar. Therefore, Gaussian Mixture Model (GMM) [28] can best describe these mixed groups.

We need to separate two Gaussian distribution groups from mixed groups and estimate the parameters of the two groups. In this way, the problem exactly conforms to the GMM. For our problem, the probability distribution model is shown as follows.

$$P(y|\theta)=\alpha_1\varphi(y|\theta_1)+\alpha_2\varphi(y|\theta_2)\alpha_1,\alpha_2\geq 0,\alpha_1+\alpha_2=1,\theta=$$

where $\varphi(y|\theta_k)$ is the density function of the Gaussian distribution $N(\mu_k,\sigma_k^2)$, which is the k-th sub-model.

The model can be trained by EM algorithm. All observation data (accounts) need to be input to output model parameters, so as to determine which sub-model each account is more likely to belong to. In this way, all accounts are split into two groups.

5 Evaluation

5.1 Dataset and Metrics

Our original dataset is shown in Table 3.

Table 3: Statistics of dataset.

Class	Users	Tweets	Crawled at
CCP	991	1610034	2014-11-14
Legit	1083	2839361	2015-05-02

Although our method is a clustering method, since our dataset has class label, we can still use the performance measures in the standard classification method, such as precision and recall. Considering CCP as positive class and legit as negative class, our evaluation metrics include Precision, Recall, Accuracy and F_β-measure, where $\beta>0$ is a weight coefficient, measuring relative importance of precision and recall. The more β is, the more relatively important the recall is [29]. In general, β is chosen as 1, denoting that precision is as important as recall. But in our case, we should try our best to avoid undetected malicious accounts, so recall is a little more important than precision. Here we choose β as 2, and F-measure become F2-measure.

5.2 Evaluation Against Baseline Methods

5.2.1 Settings. We need to simulate an early detection scenario, and hope to detect CCPs at any time. So, two date is chosen, i.e. October 30, 2014 (while-event) and October 24, 2014 (pre-event), respectively. For each of the two detection dates, the online behavior of each account can only be selected before the detection date. Therefore, two datasets are constructed, called while-event dataset (WED) and pre-event dataset (PED).

5.2.2 Baseline Methods. **Supervised content polluters classification.** We choose the detection method proposed by Caverlee's research team [30], whose dataset is collected for about seven months through Honeypot technology they developed, including of 22223 content polluters and 19276 legitimate accounts. Using the features of User Demographics, User Friendship Networks, User Content and User History, they selected random forest classifier to classify malicious accounts, and the accuracy and F1 could exceed 98%. Now, we will use their dataset to train a random forest classifier, and test its robustness on our dataset. Since the format of the two datasets does not match, we extracted the common features between them. Nevertheless, the 10-fold cross-validation on Caverlee's dataset still achieved over 96% precision and recall.

Unsupervised sequence clustering. Different from the detection of classification method one by one, the clustering method is group detection and does not require training process. Our strong baseline, Social Fingerprinting [19] belongs to sequence clustering, which has been detailed in Section 2.2.

5.2.3 Evaluation Result. The evaluation results on WED and PED are presented in Table 4 and Table 5, respectively.

Table 4: Performance comparison between baseline and CrowdGuard on while-event dataset.

Method	Precision	Recall	Accuracy	F2
Caverlee et al. [30]	0.535	**0.971**	0.583	0.835
Social Fingerprinting [19]	0.934	0.058	0.548	0.071
CrowdGuard	0.978	0.928	0.956	0.938
CrowdGuard(select features)	**0.986**	0.947	**0.968**	**0.955**

Although Caverlee's method [30] achieves the highest recall, their precision is only about 50%, and the high false alarm rate may cause trouble to many legitimate accounts. Social Fingerprinting [19] almost has no false alarm, but recall is too low, since in the early scenario, the DNA strand was too short to make the similarity of group behavior apparent. It makes a large number of content polluters go unpunished, so it is not qualified for early detection. The reason why CrowdGuard achieves the best overall performance is that not only most of our features are robust enough, but also they are bimodal, and GMM is very fit to model them. Moreover, these features combine individual and group features, and cover many aspects, especially focusing on collective behavior, which makes content polluters more difficult to evade.

Table 5: Performance comparison between baseline and CrowdGuard on pre-event dataset.

Method	Precision	Recall	Accuracy	F2
Caverlee et al. [30]	0.535	**0.976**	0.583	0.838
Social Fingerprinting [19]	0.951	0.079	0.558	0.097
CrowdGuard	0.965	0.928	0.949	0.935
CrowdGuard(select features)	**0.969**	0.940	**0.957**	**0.946**

6 Related Work

Recently, there is many academic literature on modeling and analyzing content polluters, and many detection methods have been proposed, which can be divided into individual content polluter (ICP) detection and CCP detection.

For ICP detection, we usually adopt classification method in machine learning where the key lies in feature selection and classifier selection, and feature selection is more important. Amleshwaram et al. [31] proposed 15 features of the message content to detect content polluters in Twitter. Stringhini et al. [9] used the features of friends and message content to detect spam accounts in OSN. Thomas et al. [32] proposed a real-time URL detection scheme based on the page content to which the URL points. Egele et al. [33] extract 7 content features, modeling the messages, and judge whether the messages published later deviate from the created model to detect compromised accounts. Different features have different robustness, with some easily evaded and others difficult. Yang et al. [15] made an empirical evaluation of the robustness of 24 common features and gave possible strategies for spammers to evade detection features.

As for CCP detection, we usually detect them in a group fashion. Miller et al. [34] clustered the features of users' profile and message contents, and used StreamKM++ and DenStream combined data stream clustering algorithm to cluster legitimate accounts into one class, and those outside the class were content polluters. Wang et al. [35] used HTTP request sequences of users accessing social networks for clustering. Cao et al. [36] believed that the behaviors of content polluters showed loose synchronous behaviors in social networks, and they were detected by clustering the behaviors of accounts. Viswanath et al. [37] model the behavior of legitimate accounts through PCA, and then determine whether an account is content polluter based on the degree of deviation between itself and the model. Social Fingerprinting [19] adopts k-LCS algorithm [23] to detect CCP.

7 Conclusion and Future Work

In this paper, we focus on collective content polluters in OSN. We analyze the difference of CCPs and legitimate accounts in different aspects of features and get some findings. Using them, we extract 43 individual/group features and leverage GMM for early detection. Finally, an experimental scenario simulating early detection is designed. The results show that CrowdGuard is competent for early detection.

However, there are still some limitations. As for data, although the distinction between the two groups is obvious, there is no way that content polluters can come from only one source in the real world, and legitimate group is not necessarily made up of harmless accounts. As for methodology, group detection is really a good idea, but because there are so many users in OSN, it maybe takes a lot of time and resource to process huge amounts of data.

There are several interesting directions to explore. For model, other models can be considered, such as graph mining, anomaly detection framework [38], and so on. As for methodology, the combination of one-by-one detection and group detection can be considered. As for data, in addition to CCP, we should consider other types of content polluters and simulate real-world scenarios.

8 ACKNOWLEDGMENTS

This work was supported by the National Key R&D Program of China (2017YFB1001800), the National Natural Science Foundation of China (No. 61772428,61725205).

REFERENCES

[1] Hongyu Gao, Jun Hu, Tuo Huang, Jingnan Wang, and Yan Chen. 2011. Security Issues in Online Social Networks. IEEE Internet Computing 15, 4 (July 2011), 56-63. DOI: http://dx.doi.org/10.1109/MIC.2011.50

[2] Ting-Kai Huang, Md Sazzadur Rahman, Harsha V. Madhyastha, Michalis Faloutsos, and Bruno Ribeiro. 2013. An analysis of socware cascades in online social networks. In Proceedings of the 22nd international conference on World Wide Web (WWW' 13). ACM, New York, NY, USA, 619-630. DOI: https://doi.org/10.1145/2488388.2488443

[3] Yazan Boshmaf, Dionysios Logothetis, Georgos Siganos, Jorge Lería, Jose Lorenzo, Matei Ripeanu, Konstantin Beznosov, and Hassan Halawa. 2016. Íntegro. Comput. Secur. 61, C (August 2016), 142-168. DOI: http://dx.doi.org/10.1016/j.cose.2016.05.005

[4] Xin Ruan, Zhenyu Wu, Haining Wang, and Sushil Jajodia. 2016. Profiling Online Social Behaviors for Compromised Account Detection. IEEE Trans. Information Forensics and Security, 11(1), 176-187.

[5] Abdullah Almaatouq, Erez Shmueli, Mariam Nouh, Ahmad Alabdulkareem, Vivek K. Singh, Mansour Alsaleh, Abdulrahman Alarifi, Anas Alfaris, and Alex 'Sandy' Pentland. 2016. If it looks like a spammer and behaves like a spammer, it must be a spammer: analysis and detection of microblogging spam accounts. Int. J. Inf. Secur. 15, 5 (October 2016), 475-491. DOI: http://dx.doi.org/10.1007/s10207-016-0321-5

[6] Mehwish Nasim, Andrew Nguyen, Nick Lothian, Robert Cope, and LewisMitchell. 2018. Real-time Detection of Content Polluters in Partially Observable Twitter Networks. In WWW '18 Companion: The 2018 Web Conference Companion, April 23-27, 2018, Lyon, France. ACM, New York, NY, USA, 9 pages. DOI: https://doi.org/10.1145/3184558.3191574

[7] Yazan Boshmaf, Ildar Muslukhov, Konstantin Beznosov, and Matei Ripeanu. 2011. The socialbot network: when bots socialize for fame and money. In Proceedings of the 27th Annual Computer Security Applications Conference (ACSAC '11). ACM, New York, NY, USA, 93-102. DOI: http://dx.doi.org/10.1145/2076732.2076746

[8] Muhammad Al-Qurishi, Mabrook Al-Rakhami, Atif Alamri, Majed Alrubaian, Sk Md Mizanur Rahman, and M. Shamim Hossain. 2017. Sybil Defense Techniques in Online Social Networks: A Survey. in IEEE Access, 5, 1200-1219.

[9] Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. 2010. Detecting spammers on social networks. In Proceedings of the 26th Annual Computer Security Applications Conference (ACSAC '10). ACM, New York, NY, USA, 1-9. DOI: http://dx.doi.org/10.1145/1920261.1920263

[10] Eva Zangerle and Günther Specht. 2014. "Sorry, I was hacked": a classification of compromised twitter accounts. In Proceedings of the 29th Annual ACM Symposium on Applied Computing (SAC '14). ACM, New York, NY, USA, 587-593. DOI: https://doi.org/10.1145/2554850.2554894

[11] Hongyu Gao, Jun Hu, Christo Wilson, Zhichun Li, Yan Chen, and Ben Y. Zhao. 2010. Detecting and characterizing social spam campaigns. In Proceedings of the 10th ACM SIGCOMM conference on Internet measurement (IMC '10). ACM, New York, NY, USA, 35-47. DOI: http://dx.doi.org/10.1145/1879141.1879147

[12] Emilio Ferrara, Onur Varol, Clayton Davis, Filippo Menczer, and Alessandro Flammini. 2016. The rise of social bots. Commun. ACM 59, 7 (June 2016), 96-104. DOI: https://doi.org/10.1145/2818717

[13] Tommi Vatanen, Mikael Kuusela, Eric Malmi, Tapani Raiko, Timo Aaltonen, and Yoshikazu Nagai. 2012. Semi-supervised detection of collective anomalies with an application in high energy particle physics. In IJCNN, 1-8.

[14] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. ACM Comput. Surv. 41, 3, Article 15 (July 2009), 58 pages. DOI: http://dx.doi.org/10.1145/1541880.1541882

[15] Chao Yang, Robert Harkreader, and Guofei Gu. 2013. Empirical Evaluation and New Design for Fighting Evolving Twitter Spammers," in IEEE Transactions on Information Forensics and Security, 8(8), 1280-1293.

[16] Clayton Allen Davis, Onur Varol, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer. 2016. BotOrNot: A System to Evaluate Social Bots. In Proceedings of the 25th International Conference Companion on World Wide Web (WWW '16 Companion). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 273-274. DOI: https://doi.org/10.1145/2872518.2889302

[17] Onur Varol, Emilio Ferrara, Clayton A. Davis, Filippo Menczer, and Alessandro Flammini. 2017. Online human-bot interactions: Detection, estimation, and characterization. arXiv: 1703.03107. Retrieved from https://arxiv.org/abs/1703.03107

[18] Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. 2016. DNA-inspired online behavioral modeling and its application to spambot detection. IEEE Intelligent Systems, 31(5), 58-64.

[19] Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. 2018. Social fingerprinting: detection of spambot groups through DNA-inspired behavioral modeling. IEEE Transactions on Dependable and Secure Computing, 15(4), 561-576.

[20] Zhe Zhao, Paul Resnick, and Qiaozhu Mei. 2015. Enquiring Minds: Early Detection of Rumors in Social Media from Enquiry Posts. In Proceedings of the 24th International Conference on World Wide Web (WWW '15). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 1395-1405. DOI: https://doi.org/10.1145/2736277.2741637

[21] Tu Ngoc Nguyen, Cheng Li, and Claudia Niederée. 2017. On Early-stage Debunking Rumors on Twitter: Leveraging the Wisdom of Weak Learners. arXiv: 1709.04402. Retrieved from https://arxiv.org/abs/1703.04402

[22] Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. 2017. The Paradigm-Shift of Social Spambots: Evidence, Theories, and Tools for the Arms Race. In Proceedings of the 26th International Conference on World Wide Web Companion (WWW '17 Companion). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 963-972. DOI: https://doi.org/10.1145/3041021.3055135

[23] Michael Arnold and Enno Ohlebusch. 2011. Linear Time Algorithms for Generalizations of the Longest Common Substring Problem. Algorithmica 60, 4 (August 2011), 806-818. DOI: https://doi.org/10.1007/s00453-009-9369-1

[24] Huayi Li, Geli Fei, Shuai Wang, Bing Liu, Weixiang Shao, Arjun Mukherjee, and Jidong Shao. 2017. Bimodal Distribution and Co-Bursting in Review Spam Detection. In Proceedings of the 26th International Conference on World Wide Web (WWW '17). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 1063-1072. DOI: https://doi.org/10.1145/3038912.3052582

[25] Bin Guo, Chao Chen, Daqing Zhang, Zhiwen Yu, and Alvin Chin. 2016. Mobile crowd sensing and computing: when participatory sensing meets participatory social media. in IEEE Communications Magazine, 54(2), 131-137.

[26] Bin Guo, Zhu Wang, Zhiwen Yu, Yu Wang, Neil Y. Yen, Runhe Huang, and Xingshe Zhou. 2015. Mobile Crowd Sensing and Computing: The Review of an Emerging Human-Powered Sensing Paradigm. ACM Comput. Surv. 48, 1, Article 7 (August 2015), 31 pages. DOI: https://doi.org/10.1145/2794400

[27] Yiming Yang and Jan O. Pedersen. 1997. A comparative study on feature selection in text categorization. In Icml, 97, 412-420.

[28] Jeffrey A. Bilmes. 1998. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. International Computer Science Institute, 4(510), 126.

[29] Christopher M. Bishop. 2006. Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag, Berlin, Heidelberg.

[30] Kyumin Lee, Brian David, and James Caverlee. 2011. Seven Months with the Devils: A Long-Term Study of Content Polluters on Twitter. In ICWSM, 185-192.

[31] Amit A. Amleshwaram, Narasimha Reddy, Sandeep Yadav, Guofei Gu, and Chao Yang. 2013. CATS: Characterizing automation of Twitter spammers. 2013 Fifth International Conference on Communication Systems and Networks (COMSNETS), Bangalore, 2013, 1-10.

[32] Kurt Thomas, Chris Grier, Justin Ma, Vern Paxson, and Dawn Song. 2011. Design and Evaluation of a Real-Time URL Spam Filtering Service. In Proceedings of the 2011 IEEE Symposium on Security and Privacy (SP '11). IEEE Computer Society, Washington, DC, USA, 447-462. DOI: https://doi.org/10.1109/SP.2011.25

[33] Manuel Egele, Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. 2015. Towards detecting compromised accounts on social networks. IEEE Transactions on Dependable & Secure Computing, 14(4), 447-460.

[34] Zachary Miller, Brian Dickinson, William Deitrick, Wei Hu, and Alex Hai Wang. 2014. Twitter spammer detection using data stream clustering. Information Sciences, 260(1), 64-73.

[35] Gang Wang, Tristan Konolige, Christo Wilson, Xiao Wang, Haitao Zheng, and Ben Y. Zhao. 2013. You are how you click: clickstream analysis for Sybil detection. In Proceedings of the 22nd USENIX conference on Security (SEC'13). USENIX Association, Berkeley, CA, USA, 241-256.

[36] Qiang Cao, Xiaowei Yang, Jieqi Yu, and Christopher Palow. 2014. Uncovering Large Groups of Active Malicious Accounts in Online Social Networks. In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS '14). ACM, New York, NY, USA, 477-488. DOI: https://doi.org/10.1145/2660267.2660269

[37] Bimal Viswanath, M. Ahmad Bashir, Mark Crovella, Saikat Guha, Krishna P. Gummadi, Balachander Krishnamurthy, and Alan Mislove. 2014. Towards detecting anomalous user behavior in online social networks. In Proceedings of the 23rd USENIX conference on Security Symposium (SEC'14). USENIX Association, Berkeley, CA, USA, 223-238.

[38] Amanda Minnich, Nikan Chavoshi, Danai Koutra, and Abdullah Mueen. 2017. BotWalk: Efficient Adaptive Exploration of Twitter Bot Networks. In Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017 (ASONAM '17), Jana Diesner, Elena Ferrari, and Guandong Xu (Eds.). ACM, New York, NY, USA, 467-474. DOI: https://doi.org/10.1145/3110025.3110163