CrowdTarget: Target-based Detection of Crowdturfing in Online Social Networks

Jonghyuk Song
Dept. of CSE, POSTECH
Pohang, Republic of Korea
freestar@postech.ac.kr

Sangho Lee
Dept. of CSE, POSTECH
Pohang, Republic of Korea
sangho2@postech.ac.kr

Jong Kim
Dept. of CSE, POSTECH
Pohang, Republic of Korea
jkim@postech.ac.kr

Abstract

Malicious crowdsourcing, also known as crowdturfing, has become an important security problem. However, detecting accounts performing crowdturfing tasks is challenging because human workers manage the crowdturfing accounts such that their characteristics are similar with the characteristics of normal accounts. In this paper, we propose a novel crowdturfing detection method, called *CrowdTarget*, that aims to detect target objects of crowdturfing tasks (e.g., post, page, and URL) not accounts performing the tasks. We identify that the manipulation patterns of target objects by crowdturfing workers are unique features to distinguish them from normal objects. We apply CrowdTarget to detect collusion-based crowdturfing services to manipulate account popularity on Twitter with artificial retweets. Evaluation results show that CrowdTarget can accurately distinguish tweets receiving crowdturfing retweets from normal tweets. When we fix the false-positive rate at 0.01, the best true-positive rate is up to 0.98.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*; K.4.1 [**Computers and Society**]: Public Policy Issues—*Abuse and crime involving computers*

General Terms

Security

Keywords

Malicious crowdsourcing; Online social networks; Twitter; Underground services

1. INTRODUCTION

According to the characteristics of tasks, people can do certain tasks better than computers in terms of accuracy, cost, and speed. *Crowdsourcing* is the process of outsourcing tasks to *human workers* to exploit such observations while paying them for the tasks. Various crowdsourcing sites exist, such as Amazon Mechanical Turk, Microworkers, and Crowdsource.

Unfortunately, adversaries have become major customers of crowdsourcing services. They use the services for malicious purposes because human workers can easily circumvent conventional security systems to detect automated activities performed by *bots*. Adversaries can leave various malicious tasks to human workers belonging to crowdsourcing sites, such as spreading spam URLs, searching specific keywords to manipulate search results, and boosting the popularity of their accounts in online social networks (OSNs). This malicious crowdsourcing has both characteristics of crowdsourcing and astroturfing, so researchers name it *crowdturfing* [33].

Although researchers propose numerous methods of malicious account detection using *account-based features* or *synchronized group activities*, they are inappropriate to detect crowdturfing accounts. First, detection methods based on account-based features [12, 17, 23, 34, 35] inspect the characteristic of individual account, e.g., the number of friends, the number of posts, and age. However, recent studies [29, 32] show that applying the techniques to detect crowdturfing accounts is vulnerable to simple evasion techniques, such as performing malicious tasks while doing normal behaviors. Interestingly, our analysis of account popularity, which is computed by using account features and behaviors, shows that crowdturfing accounts are more popular than normal accounts (Section 4.1).

Next, identifying synchronized group activities of malicious accounts is state-of-the-art methods of detecting malicious accounts managed by bots [8, 11, 15, 16, 31]. However, we empirically identify that crowdturfing tasks have weak correlation because human workers perform the tasks either without schedule or with flexible schedule (Section 4.2). Consequently, we demand a novel detection method that relies on neither account characteristics nor program-controlled behaviors.

In this paper, we propose a novel method of detecting crowdturfing, called *CrowdTarget*. CrowdTarget aims to discover *target objects* that crowdturfing customers attempt to manipulate, e.g., URL, search keyword, and post, by using their *manipulation patterns*. Unlike conventional detection methods using account characteristics, CrowdTarget is (i) robust against evasive techniques to manipulate account-based features. Also, it can detect crowdturfing tasks per-

CCS'15, October 12–16, 2015, Denver, Colorado, USA.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3832-5/15/10 ...$15.00.
DOI: http://dx.doi.org/10.1145/2810103.2813661.

formed by (ii) new accounts or (iii) casual workers who occasionally participate in crowdturfing tasks.

Among numerous crowdturfing services aiming at various services, we apply CrowdTarget to *collusion-based crowdturfing services* that manipulate account popularity on Twitter by using artificial retweets. Our goal is to distinguish between tweets receiving retweets from crowdturfing accounts (we name them *crowdturfing tweets*) and tweets receiving retweets from normal accounts.

We first analyze the differences in retweet patterns of the three tweet groups: normal, crowdturfing, and black-market tweet groups. From the analysis, we find four new retweet-based features that allow us to distinguish crowdturfing tweets from others: (i) retweet time distribution, (ii) the ratio of the most dominant application, (iii) the number of unreachable retweeters, and (iv) the number of received clicks. The first feature, retweet time distribution, consists of four sub-features: mean, standard deviation, skewness, and kurtosis.

Next, we build three classification models, Ada Boost, Gaussian naïve Bayes, and k-nearest neighbors, by using the retweet-based features and evaluate them with our ground-truth dataset. Evaluation results show that CrowdTarget can accurately distinguish crowdturfing tweets from normal tweets; the true-positive rate (TPR) is 0.98 when the false-positive rate (FPR) is 0.01 with the k-nearest neighbor algorithm.

In summary, the main contributions of this paper are as follows:

- **New detection approach.** We detect crowdturfing by analyzing not the characteristics of its accounts but the characteristics of its targets. In this paper, the targets are tweets and the crowdturfing task retweets the tweets. To the best of our knowledge, this is the first approach that detects crowdturfing by using the targets.
- **In-depth analysis.** We analyze retweets generated by three account groups: normal, crowdturfing, and black market. This analysis provides insight to understand each group's behaviors.
- **High accuracy.** The accuracy of our method is very high. When we fix the false-positive rate at 0.01, the true-positive rate is up to 0.98.

The remainder of this paper is organized as follows. In Section 2 we compare black-market sites and crowdturfing sites. In Section 3 we explain the details of our dataset. In Section 4 we analyze the characteristics of crowdturfing workers. In Section 5 we introduce the unique features of crowdturfing targets. In Section 6 we explain how we use the features to construct our classifiers and evaluate their accuracy. In Section 7 we discuss the robustness of our features. In Section 8 we introduce related studies. Lastly, we conclude this paper in Section 9.

2. BACKGROUND

In this section, we explain black-market sites and crowdturfing sites for OSNs. Their main difference is that the black-market sites only sell malicious services, whereas the crowdturfing sites not only sell malicious services but also encourage the participation of users in conducting malicious activities.

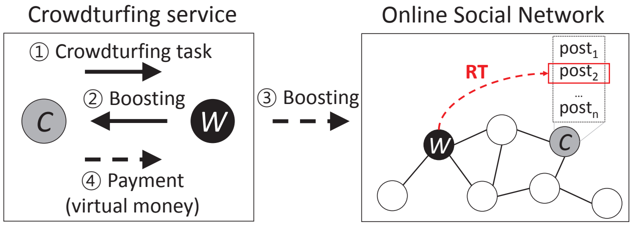


Figure 1: Procedure of OSN boosting in a collusion-based crowdturfing service. A customer C posts a task on the service S. A worker W performs the task on S and S relays W's actions to the target OSN. C finally pays virtual money for the tasks that W has conducted.

2.1 Black-market Site for OSNs

Black-market sites are proposed to satisfy people's desire: promoting their popularity in OSNs. The sites provide various services for the goal, e.g., increasing the number of followers, likes, and comments. According to the price, they offer various plans with deadlines, e.g., $39 for gaining 2,500 Twitter followers within 48 hours.

To provide malicious services, black-market sites usually operate a large number of bots to perform many tasks by deadlines. They strive to develop bot accounts that closely resemble normal accounts because (i) they want to prevent security teams of OSNs from suspending their accounts and (ii) their customers want to have human-like followers to make the popularity of their accounts more realistic.

Although bot accounts resemble normal accounts, they inevitably have synchronized group activities because they should perform the same tasks by deadlines. Therefore, recent studies try to detect bot accounts in OSNs by discovering their synchronized group activities [8,11,15,16,31]. In Section 4, we also observe synchronized group activities of black-market accounts.

2.2 Crowdturfing Sites for OSNs

Recently, *collusion-based crowdturfing services* specialized for OSN boosting have appeared, e.g., `addmefast.com` [1] and `traffup.net` [6]. In these services, users *exchange their efforts* to achieve their goals, such as increasing the number of Twitter followers and retweets, the number of Instagram comments, and the number of Facebook likes. Figure 1 shows the procedure of OSN boosting in such services.

① A customer C posts an object (e.g., tweet and page) to be manipulated to a crowdturfing service and specifies a reward (e.g., an amount of virtual money).

② A worker W performs boosting tasks on the crowdturfing service (e.g., click an RT button).

③ The crowdturfing service relays the boosting tasks to the target OSN.

④ The crowdturfing service transfers C's virtual money to W.

The collusion-based crowdturfing service simplifies the process of boosting for both workers and customers. In a con-

Table 1: The dataset

Dataset	#Tweets	#Retweets	#Retweeters
Normal			
Without URL	10,318	914,974	390,275
With URL	15,248	1,941,482	1,149,563
Total	25,566	2,856,456	1,412,632
Crowdturfing			
Without URL	4,531	576,033	115,657
With URL	14,867	1,866,843	110,295
Total	19,398	2,442,876	190,800
Black-market			
Total	282	71,858	41,829

ventional crowdsourcing service, a worker performs the boosting in the target OSN and customers should examine whether the worker has done the task properly. However, the collusion-based crowdturfing service automates the procedures of workers and customers. When a user signs up the crowdturfing service, the user authorizes the crowdturfing service's application that manages and monitors overall boosting tasks. The application monitors how crowdturfing workers perform certain boosting tasks at the crowdturfing service and relays the tasks to the target OSN. Thus, the service can be convinced that the boosting tasks are done properly. This convenient procedure makes workers easily perform many crowdturfing tasks.

Based on the analysis results in Section 4, we are convinced that workers of collusion-based crowdturfing services are either real humans or advanced human-like bots. Unlike casual bots, the crowdturfing workers are more popular than normal accounts and do not have synchronized group activities. Therefore, the conventional bot detectors cannot detect the crowdturfing accounts.

3. DATA COLLECTION

In this section, we explain our ground-truth tweets collected from three sources: Twitter, crowdturfing sites, and black-market sites. We only consider tweets that received $\geq$ 50 retweets because a small number of retweets cannot manipulate the popularity of accounts. Note that every black-market site analyzed assures $\geq$ 50 retweets, so this treatment is acceptable. Also, every tweet we collected was created between November 2014 and February 2015. Table 1 summarizes our dataset.

3.1 Ground-truth Tweets

Normal tweets on Twitter. We collected normal tweets from Twitter. We regarded a tweet as a normal tweet if it was created by a verified Twitter account that has $\geq$ 100,000 followers. We randomly selected 1,044 verified Twitter accounts that satisfy the requirements and monitored their timeline to collect tweets and retweets.

Crowdturfing tweets. We collected crowdturfing tweets from nine different crowdturfing sites. We registered at the crowdturfing sites and retrieved tasks requesting retweets posted on the sites.

Black-market tweets. We collected black-market tweets from five different black-market sites, e.g., `retweets.pro` [4] and `socialshop.co` [5]. We first wrote 282 tweets containing URLs by using our fake Twitter accounts. Then, we registered at the black-market sites and purchased retweets for our tweets. On average, we paid \$5.6 for 100 retweets and \$13.4 for 1,000 retweets. All black-market sites provided the retweets about a day.

3.2 Methods to Collect Retweets

We explain our approach to collect retweets. Although Twitter provides a REST API to retrieve retweets that a tweet received (`statuses/retweets`), this API only returns up to 100 latest retweets. Our objective is to collect as many retweets for each target tweet as possible. We take two approaches to achieve it. First, for a target tweet recently posted, we use a streaming API to monitor retweets it will receive in the next three days. Second, for a target tweet posted in the past, we use a Twitter search function to find as many retweets of the target tweet as possible.

3.3 Ethics

In this study, we have encountered several legal and ethical problems on experimenting and collecting data. We referenced Thomas *et al.* [25]'s approach to ethically study underground services. We designed our data collection and subsequent experiments to follow the exemption guideline from a formal review of the institutional review board (IRB) of our institute.

First, we have not collected any data that can be used to distinguish individual subjects. We deleted detailed personal information (e.g., names and profiles) that were unrelated to our experiments.

Second, to minimize our effects to underground services, we only retrieved public tasks posted on crowdturfing sites and purchased a small number of retweets from black-market sites. Further, we neither attempted to inspect who operate such services nor contacted them via other channels.

Third, to avoid the negative effects of using black-market services against Twitter and its users, we deleted our fake accounts right after receiving and collecting purchased retweets. Since we made our fake accounts only follow each other and post tweets with harmless and meaningless contents, legitimate users would rarely see or retweet our tweets.

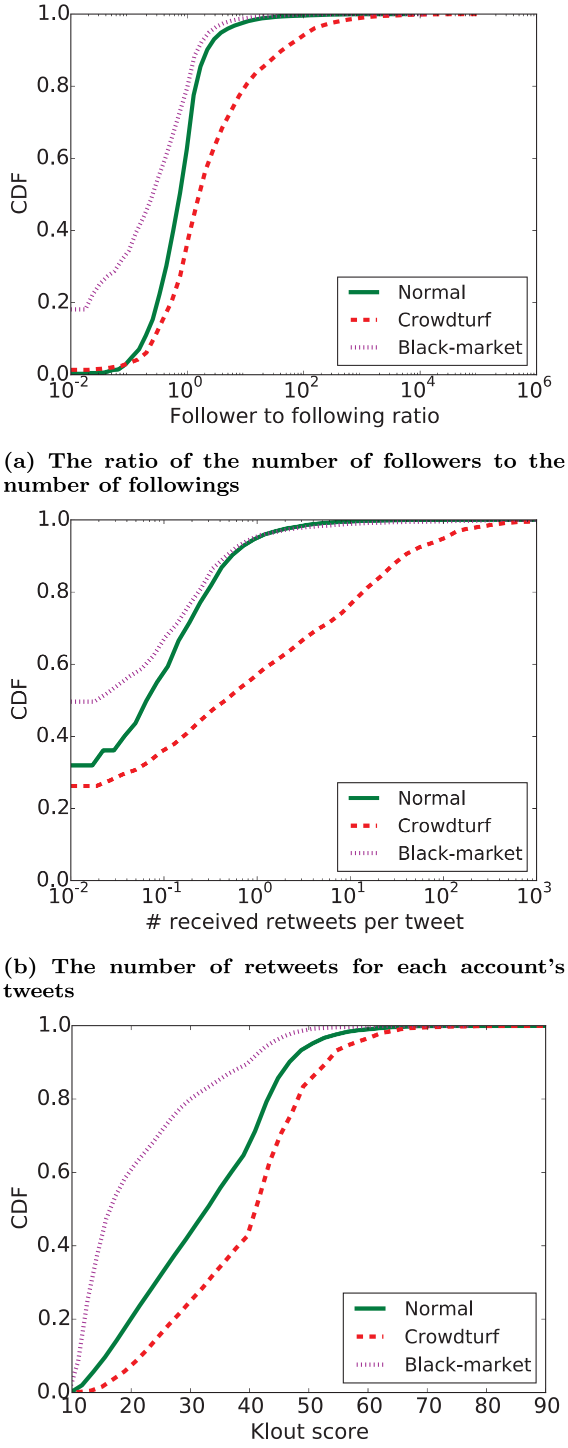
4. CROWDTURFING WORKERS

In this section, we analyze crowdturfing workers to know whether they are humans, bots, or something else. We check two sets of features: account popularity and synchronized group activity.

4.1 Account Popularity

We compare the popularity of crowdturfing accounts on Twitter with other account groups by using three features: follower to following ratio, the number of received retweets per tweet, and Klout score [3] (Figure 2). First, we measure the ratio of the number of followers to the number of followings in each account group. Figure 2a shows that approximately 70% of the crowdturfing accounts have a larger number of followers than followings; this ratio is much higher than the normal (37%) and black-market account groups (20%).

Second, we check the tweets of each account group to know how many times they are retweeted (Figure 2b). We observe that tweets posted by crowdturfing accounts are more frequently retweeted than tweets posted by normal or black-market account groups. Approximately 43% of



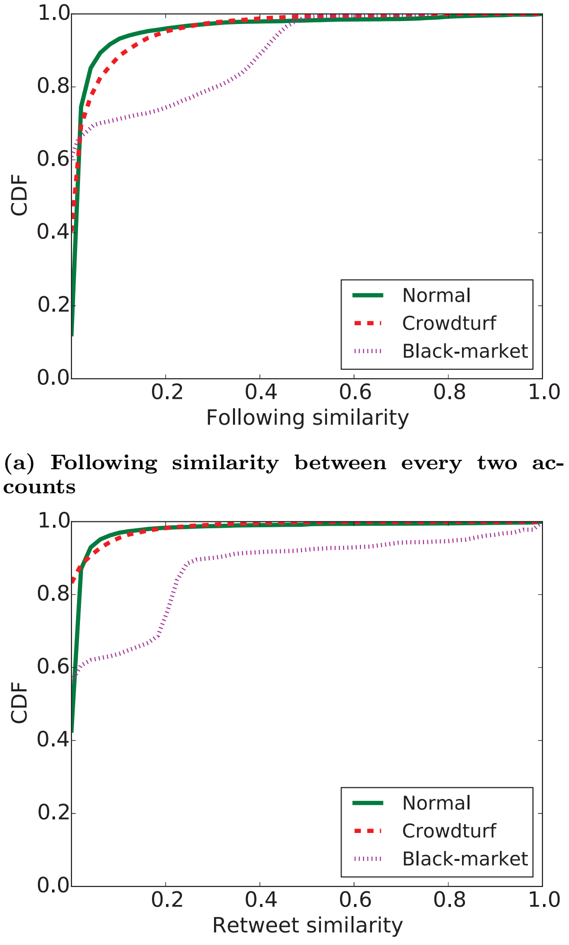
(a) The ratio of the number of followers to the number of followings

(b) The number of retweets for each account's tweets

(c) Klout score

Figure 2: Social popularities of the three account groups: normal, crowdturfing, and black-market account groups. Crowdturfing accounts are more popular than normal and black-market accounts.

tweets posted by crowdturfing accounts are retweeted more than once. In contrast, 5% of tweets posted by normal accounts and 4% of tweets posted by black-market accounts are retweeted more than once.



(a) Following similarity between every two accounts

(b) Retweet similarity between every two accounts

Figure 3: Synchronized group activities of the three account groups: normal, crowdturfing, and black-market account groups. Crowdturfing and normal accounts have similar patterns.

Third, we query Klout scores of the three account groups, which is a popular OSN influence score. Figure 2c shows that crowdturfing accounts have a higher Klout score than those of other groups. The median Klout score of the crowdturfing accounts is 41. In contrast, the median Klout scores of the normal accounts and black-market accounts are 33 and 20, respectively.

Consequently, we are convinced that crowdturfing accounts successfully boost their popularity by gaining followers and retweets from crowdturfing services. They differ from black-market accounts and resemble influential users in OSNs.

4.2 Synchronized Group Activity

Next, we aim to identify whether crowdturfing accounts have synchronized group activities. We use two measures to check it: *following similarity* and *retweet similarity*.

Following similarity. We define the following similarity F_{sim} between two accounts u_i and u_j as follows:

$$F_{sim}(u_i, u_j) = \frac{|F(u_i) \cap F(u_j)|}{|F(u_i) \cup F(u_j)|},$$

where $F(u_i)$ is a set of u_i's followings. We compute the following similarity between two accounts only when at least one of their retweets originate from the same tweets. Figure 3a shows that the crowdturfing and normal account groups have the same pattern: low following similarities. In contrast, the black-market account group has the highest following similarity.

Retweet similarity. To compute the retweet similarity, we first define a set of retweets of u_i, $RT(u_i)$, as follows:

$$RT(u_i) = \{(u_i, T_1, tid_1), (u_i, T_2, tid_2), \ldots, (u_i, T_n, tid_n)\},$$

where T_i represents retweet time and tid_i is the ID of a tweet retweeted by u_i. A retweet (u_i, T_k, tid_k) in $RT(u_i)$ is *matched* with another retweet (u_j, T_l, tid_l) in $RT(u_j)$ if they satisfy the following two properties:

1. The two retweets are for the same tweet: $tid_k = tid_l$.

2. The two retweets are created within a threshold time window: $|T_k - T_l| \leq T_{threshold}$.

Based on the definitions, we compute the retweet similarity RT_{sim} between two accounts u_i and u_j as follows:

$$RT_{sim}(u_i, u_j) = \frac{|RT(u_i) \cap RT(u_j)|}{|RT(u_i) \cup RT(u_j)|}.$$

Figure 3b shows the statistics of the retweet similarities of the three account groups. We observe that the crowdturfing and normal account groups have the same pattern: low retweet similarities. In contrast, the black-market account group has the highest retweet similarity.

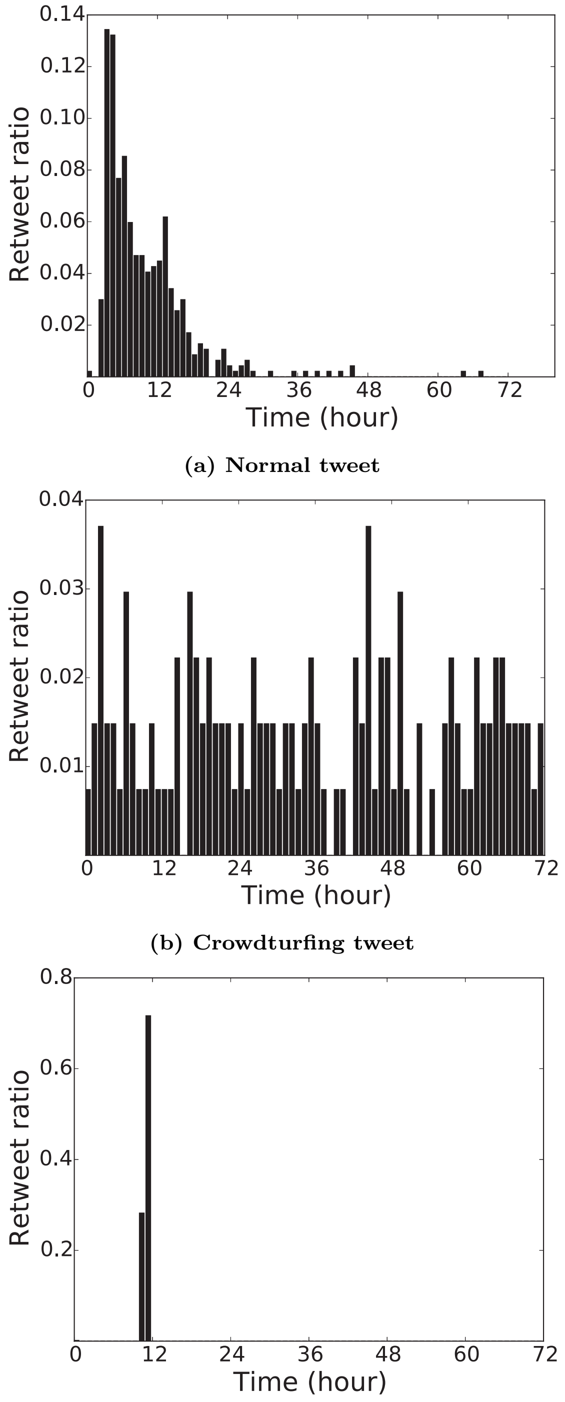
Consequently, we confirm that the crowdturfing account group shows no or weakly synchronized group activity. Thus, we should not rely on conventional detection methods using synchronized group activity to detect them.

5. CROWDTURFING TARGETS

In this section, we analyze the characteristics of crowdturfing targets on Twitter: tweets receiving artificial retweets generated by crowdturfing workers. Note that all characteristics explained in this section were never considered in previous work and we will use all of them to build our classifiers explained in Section 6.

5.1 Retweet Time Distribution

We first consider the time distribution of retweets that a tweet received. Our key insight is that the time pattern of artificial retweets differs from that of normal retweets. Figure 4 shows example retweet time distributions of normal, crowdturfing, and black-market tweets. We have counted the number of retweets generated every hour from the creation of the individual tweets. Figure 4a shows that the normal tweet is intensively retweeted within a few hours after posting, and number of retweets decreases as time goes on. In contrast, Figure 4b shows that the crowdturfing tweet is constantly retweeted because the tweet is continuously exposed to crowdturfing workers as long as it is posted on crowdturfing services. In the black-market case (Figure 4c), a large number of retweets are generated within a certain time period, and no other retweet is generated later.



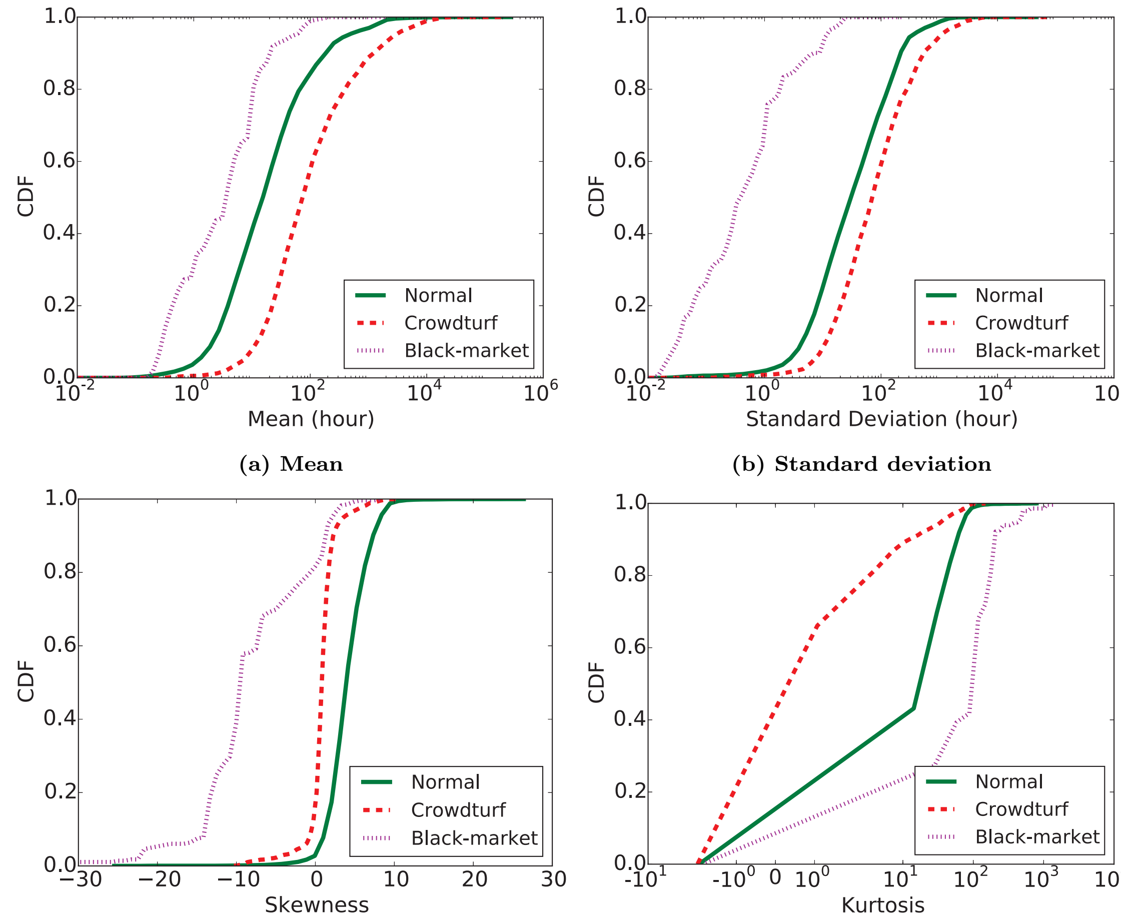
(a) Normal tweet

(b) Crowdturfing tweet

(c) Black-market tweet

Figure 4: Retweet time distributions of normal, crowdturfing, and black-market tweets. They differ from each other.

To extract features from the retweet time distribution, we use four popular measures to figure out the shape of a distribution.



(a) Mean

(b) Standard deviation

(c) Skewness

(d) Kurtosis

Figure 5: Statistical characteristics of retweet time distribution. The characteristics of normal, crowdturfing, and black-market retweets differ from each other.

5.1.1 Mean

We use the mean of a retweet time distribution to know the average time difference between posting and retweeting. The mean retweet time of a normal tweet is usually smaller than that of a crowdturfing tweet. The mean retweet time of a black-market tweet depends on when bots begin to operate; usually, they perform retweets as soon as possible to satisfy their customers.

Figure 5a shows the mean retweet time of the three groups. The mean retweet time of the crowdturfing tweets is larger than other groups, since they are continuously retweeted. Also, approximately 90% of the black-market tweets, 60% of the normal tweets, and 20% of the crowdturfing tweets receive 50% of their retweets within 24 hours. Therefore, we decide to use the mean of a retweet time distribution as a feature.

5.1.2 Standard deviation

We use the standard deviation of a retweet time distribution to know how many retweets are generated around the mean time. Crowdturfing retweets are evenly distributed such that their standard deviation is larger than those of normal and black-market tweets.

Figure 5b shows the standard deviation of the retweet time distribution of the three groups. The crowdturfing tweets have higher standard deviation than other groups. Further, the smallest standard deviation of the black-market tweets shows that most of them are retweeted around the mean time. Therefore, we decide to use the standard deviation of a retweet time distribution as a feature.

5.1.3 Skewness

We use the skewness of a retweet time distribution to know when a tweet is mostly retweeted. Skewness is a measure of the asymmetry of the distribution. Positive skewness means that the right side tail of the distribution is longer than the left side. In contrast, negative skewness means that the tail on the left side is longer than the right side.

Figure 5c shows the skewness of retweet time distributions of the three groups. Most of the crowdturfing tweets have near-zero skewness, which implies that they are evenly retweeted. In contrast, the skewness of the normal tweets is larger than zero, which implies that the number of retweets they receive gradually decreases as time goes on. Skewness of black-market tweets depends on how the black-market services operate their bots. Most of the black-market tweets collected have negative skewness, implying that the number of retweets gradually increases at first, but suddenly decreases later. Thus, we decide to use the skewness of a retweet time distribution as a feature.

5.1.4 *Kurtosis*

We use the kurtosis of a retweet time distribution to know the intensity of retweets within a short time period. Kurtosis is a measure of the peakedness of the distribution. If a distribution is sharper than the normal distribution, its kurtosis is positive. In contrast, if a distribution is flatter than the normal distribution, its kurtosis is negative. Note that the kurtosis of the normal distribution is zero.

Figure 5d shows that the crowdturfing tweets have the lowest kurtosis among the three groups, i.e., their retweets are evenly distributed. The kurtosis of the normal tweets is much higher than that of the crowdturfing tweets because, usually, a normal retweet time distribution has a peak around the posting time. The black-market tweets have the highest kurtosis because black-market services should generate a number of retweets within a given deadline [11]. Consequently, we decide to use the kurtosis of a retweet time distribution as a feature.

5.2 Twitter Application

We find that most of the collusion-based crowdturfing services have third-party Twitter applications to generate retweets. Their web sites provide custom interfaces for workers to easily create retweets for tweets of crowdturfing customers. Therefore, for each tweet receiving retweets, we compute the ratio of the number of the retweets generated by the most dominant application to the total number of retweets.

Figure 6 shows the ratio distributions of the dominant applications used to generate retweets. We found that dominant applications generated approximately 90% of the crowdturfing retweets and approximately 99% of the black-market retweets on average. In contrast, dominant applications generated approximately 40% of the normal retweets on average. Therefore, the ratio of the dominant applications can be a feature of crowdturfing tweets.

5.3 Unreachable Retweeter

We observe that most retweeters of a crowdturfing tweet do not follow the user who posts the tweet because crowdturfing services promote the tweet to unspecified individuals without considering their friendships on Twitter. But, in general, a tweet is propagated between users who are connected with each other on Twitter. Thus, retweeters are usually connected to a posting user by follower-following relationships.

To attest the observation, we measure how many retweeters are unreachable to posting users on Twitter. Figure 7 shows that approximately 80% of the crowdturfing tweets have over 80% of unreachable retweeters. In contrast, less than 10% of normal tweets have over 80% of unreachable retweeters. Hence, the ratio of the unreachable retweeters is another feature of crowdturfing tweets.

5.4 Click Information

One of the main purpose of malicious accounts in OSNs is spreading links to many OSN users to promote their websites or spread malwares. When malicious accounts post tweets with malicious links, they abnormally boost the tweets to expose the links to as many users as possible. Thus, detecting URL tweets retweeted by crowdturfing services is an important problem.

Our hypothesis is that when retweeting tweets that contain links, *crowdturfing accounts are not willing to click on the links* because it is not their duty. Therefore, even if a tweet with a link is heavily retweeted by such services, the number of clicks that the link receives could be small.

To confirm our hypothesis, we should measure how many times a link in a tweet is clicked on. Fortunately, many Twitter users use URL shortening services (e.g., `bit.ly` and `goo.gl`) to share URLs via Twitter and the services provide the click analytics for each shortened URL [22]. This allows us to count the number of clicks that each link receives.

We extract tweets that contain `bit.ly` and `goo.gl` shortened URLs from our dataset: 6,024 normal tweets, 3,093 crowdturfing tweets, and 282 black-market tweets (when we purchased retweets from black markets, all our tweets contained shortened URLs.) We crawl the click analytics of each shortened URL and extract the number of clicks via Twitter according to the referrer information.

Figure 8 shows the ratio of the number of clicks to the number of retweets per tweet. Over 80% of links in the normal tweets receive a larger number of clicks than the number of retweets. However, approximately 90% of links in the crowdturfing tweets receive a smaller number of clicks than the number of retweets. Furthermore, most of the links in the black-market tweets are never clicked on. From the results, we confirm that most crowdturfing and black-market accounts perform retweets without clicking on contained links because they have no reason to visit the links to retweet them. Therefore, we use the click information as the final feature of crowdturfing tweets.

6. DETECTION OF CROWDTURFING TARGETS

In this section, we explain how we build our classifiers, CrowdTarget, to detect crowdturfing targets and evaluate their accuracy. We treat both crowdturfing tweets and black-market tweets as malicious tweets and attempt to distinguish them from normal tweets.

6.1 Building Classifiers

We first explain how we prepared training and testing data using the dataset in Section 3. Note that in real-world services, the number of malicious messages is fairly smaller than the number of normal messages. For example, Twitter has announced that the portion of spam tweets is approximately 1% of the total tweets [26]. Therefore, we decided to set the ratio of malicious tweets as 1% of the total tweets. We *oversampled* normal tweets to satisfy the requirement. I.e., we randomly duplicated normal tweets until their number became 99 times larger than the number of malicious tweets.

We built classifiers by using the seven features of retweets explained in Section 5: (i) mean, (ii) standard deviation, (iii) skewness, and (iv) kurtosis of retweet time distribution, (v) the ratio of dominant applications used for retweets, (vi) the ratio of unreachable retweeters, and (vii) the ratio of the number of clicks to the number of retweets for tweets containing URLs. We normalized all feature values to be lie between 0 and 1. With these features, we tested several classifiers provided by the `scikit-learn` library (a

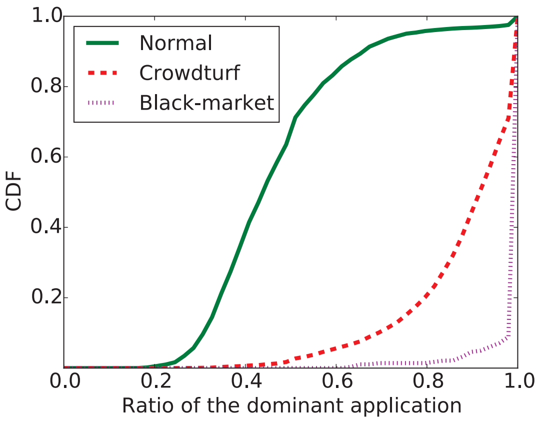


Figure 6: Ratio of the most dominant application performing retweets. Almost the same applications generate crowdturfing and black-market retweets unlike normal retweets.

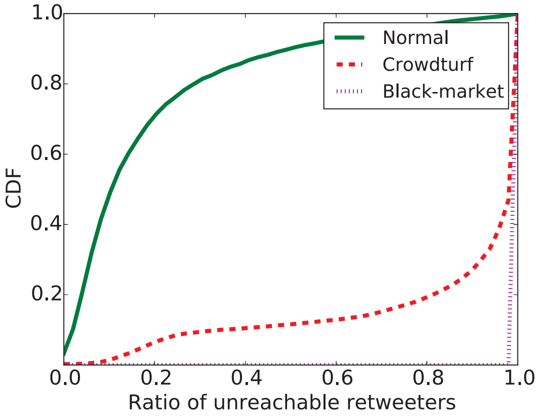


Figure 7: Ratio of unreachable retweeters per tweet. Most crowdturfing and black-market retweets are generated by unreachable retweeters who do not follow the posting users.

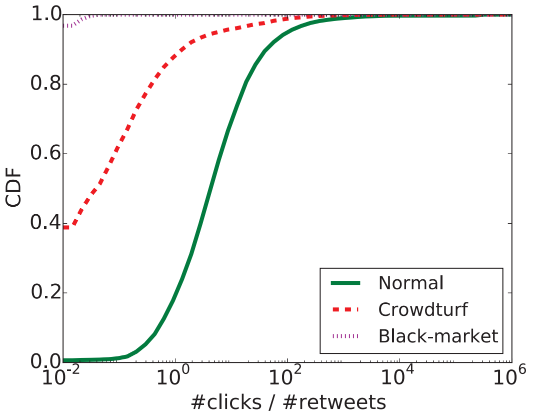


Figure 8: Ratio of the number of clicks to the number of retweets per tweet. Unlike normal retweeters, crowdturfing and black-market retweeters do not click the URLs included in the retweeted tweets.

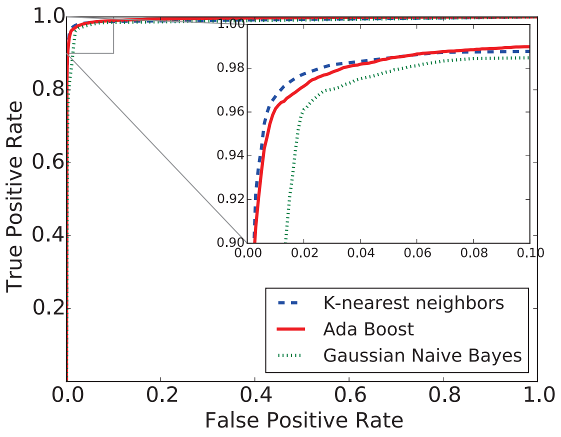


Figure 9: ROC curve showing TPRs and FPRs of CrowdTarget. We test Ada boost, Gaussian Naïve Bayes, and k-nearest neighbors algorithms with 10-fold cross validation.

Python machine-learning library) [21] and then selected top three classifiers showing good accuracy: Ada Boost, Gaussian naïve Bayes, and k-nearest neighbors. We validated classification results with 10-fold cross-validation.

6.2 Basic Classification

First, we distinguish malicious tweets from normal tweets without using click information to deal with both tweets with and without URLs. Figure 9 shows receiver operating characteristics (ROC) curves of the algorithms that draw how TPRs change according to the changes of FPRs. We define TPR and FPR are as follows:

$$TPR = \frac{\#TP}{\#TP + \#FN} \quad \text{and} \quad FPR = \frac{\#FP}{\#FP + \#TN},$$

where TP stands for true positive and FP stands for false positive.

We aim to build a classifier whose target FPR is 0.01 while increasing TPR as high as possible. When the FPR was 0.01, the TPR of the k-nearest neighbors algorithm was 0.96, the TPR of the Ada Boost algorithm was 0.95, and the TPR of the Gaussian naïve Bayes algorithm was 0.87. Therefore, we selected the k-nearest neighbors algorithm as our classifier.

We also measured the area under the ROC curve (AUC) values of the three algorithms. The AUC of the Ada Boost algorithm was 0.994, the AUC of the k-nearest neighbors algorithm was 0.991, and the AUC of the Gaussian naïve Bayes algorithm was 0.99.

6.3 Classification with Click Information

Next, we distinguish the malicious tweets containing URLs from the normal tweets containing URLs by additionally considering how many times the URLs are clicked on. We extracted tweets containing `bit.ly` and `goo.gl` links from our dataset. Then, we classified them with a link-based feature: the ratio of the number of clicks to the number of retweets. Since the k-nearest neighbors algorithm showed the best results in Section 6.2, we only tested the algorithm in this experiment for simplicity.

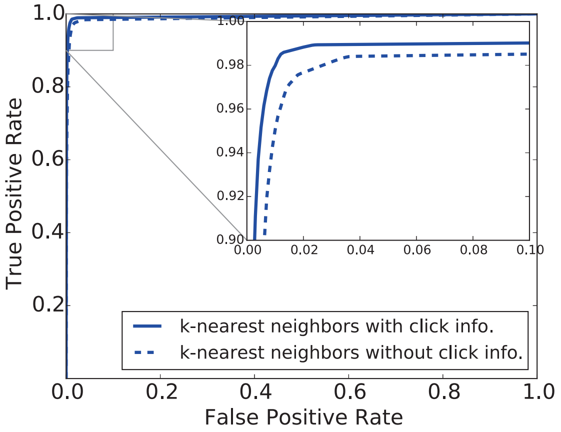


Figure 10: ROC curve showing TPRs and FPRs of CrowdTarget in distinguishing with click information and without click information. We only test k-nearest neighbors algorithm with 10-fold cross validation.

Figure 10 compares the classification results with and without click information. CrowdTarget increased accuracy by additionally considering click information. The TPR increased from 0.95 to 0.98 at FPR of 0.01, and the AUC increased from 0.989 to 0.993. Therefore, we conclude that the click information is useful to detect the malicious tweets with links.

The main shortcoming of this evaluation is that we cannot check other links that do not associated with `bit.ly` and `goo.gl` because we have no mechanism to obtain their click information. We can solve the problem if we can access the click information of `t.co` links in future (Section 7.4).

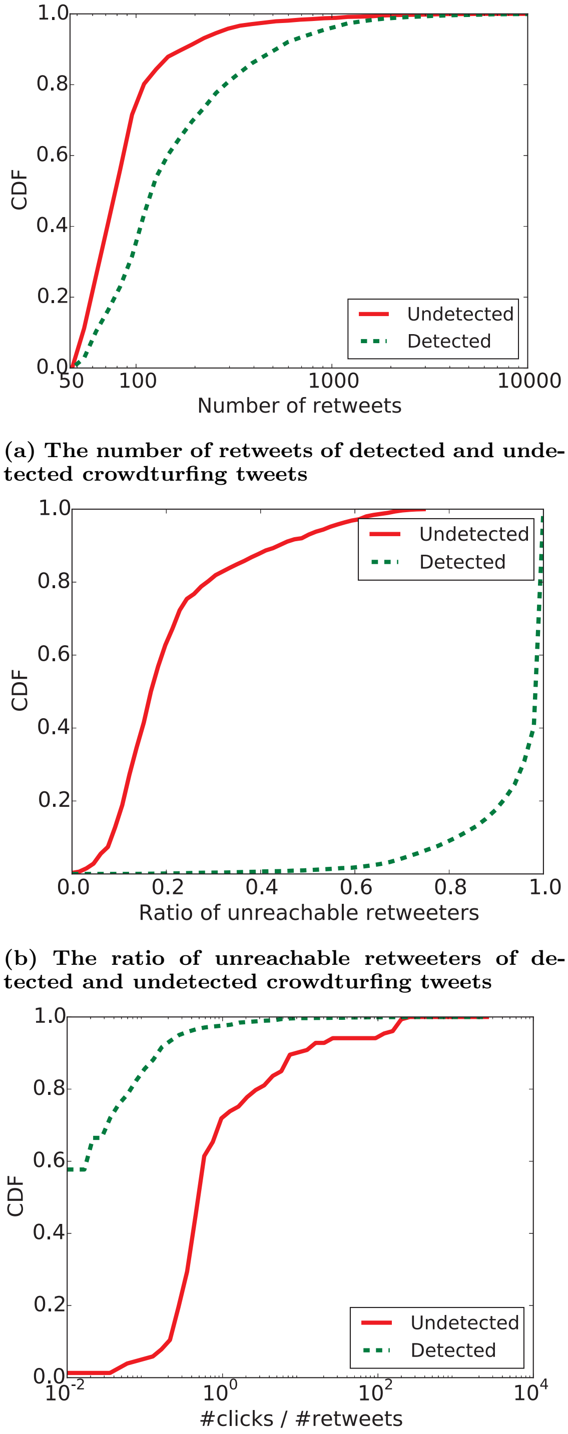
6.4 Error Analysis

In this section, we analyze the reasons of false negatives and false positives.

6.4.1 False-negative analysis

We analyzed the malicious tweets that CrowdTarget could not detect (i.e., false negatives) and figured out the following three reasons. First, we observed that CrowdTarget misjudged certain crowdturfing tweets that received a small number of retweets. Figure 11a compares the number of retweets of the detected crowdturfing tweets and that of the undetected crowdturfing tweets. The undetected crowdturfing tweets had a smaller number of retweets than that of the detected crowdturfing tweets. Approximately 75% of the undetected tweets were retweeted less than 100 times. Although CrowdTarget cannot detect crowdturfing tweets with a small number of retweets, it is not a serious problem because their negative effects against normal Twitter users are limited.

Next, we discovered that the ratio of unreachable retweeters led to more errors than other features in CrowdTarget. Figure 11b shows that approximately 50% of the undetected crowdturfing tweets were mostly retweeted by reachable accounts; the ratio of unreachable retweeters were approximately 17%. We expect that the posting users of such undetected crowdturfing tweets bought followers on the same crowdturfing service, so that their tweets will be frequently retweeted by shared followers.



(a) The number of retweets of detected and undetected crowdturfing tweets

(b) The ratio of unreachable retweeters of detected and undetected crowdturfing tweets

(c) The click ratio of detected and undetected crowdturfing tweets

Figure 11: Comparisons between detected and undetected crowdturfing tweets

Lastly, on the analysis of false negatives in the classification with click information, we recognized that a few links in the undetected crowdturfing tweets receive a larger number of clicks than retweets (Figure 11c). We searched those links on Twitter and found that they were distributed via many other tweets. Therefore, we expect that the number of clicks we measured is the aggregated number of clicks originated from every tweet containing the same links. Unfortunately, we cannot differentiate the number of clicks per tweet because `bit.ly` and `goo.gl` APIs only return domain name when retrieving referrer information (e.g., `t.co` and `twitter.com`). If we can access private data of `bit.ly`, `goo.gl`, or Twitter, we can exclude clicks from other tweets such that we can decrease the false-negative rate of CrowdTarget.

6.4.2 False-positive analysis

We manually analyzed the normal tweets classified as malicious by CrowdTarget (i.e., false positives). Most of the false positives are due to automated applications or *embedded tweets* [2].

First, we found that tweets of a few verified accounts were retweeted by automated applications. Table 2 shows examples of verified accounts that received retweets from the automated applications. We visited homepages of the applications to know their purposes and identified that they are automatic retweet applications. For example, TweetAdder is a famous automated application that was sued by Twitter due to its creation of many spam tweets [7]. Therefore, in fact, these are not false positives.

Second, CrowdTarget classified the embedded tweets in websites as malicious. Twitter offer an application, "Twitter Web Client", to allow a user to embed his or her tweets into a website. Any visitors of the website can retweet embedded tweets. However, we cannot guarantee that the visitors who have retweeted the embedded tweets are the user's followers. Consequently, the ratio of unreachable retweeters of embedded tweets is higher than normal tweets such that they can be misclassified. We think that if we can access the private date of Twitter, e.g., IP addresses of retweeters, we can avoid this problem.

7. FEATURE ROBUSTNESS

In this section, we discuss the robustness of our features against feature fabrication attempts.

7.1 Retweet Time Distribution

Retweeters can cooperate each other to artificially manipulate retweet time distributions. For the goal, they should arrange a retweet time schedule similar with a normal retweet time distribution and perform retweets as scheduled. However, it is difficult to do that by themselves because crowdturfing workers act independently.

The crowdturfing services also can attempt to manipulate the retweet time distributions. First, the services can manipulate every boosting task of a worker by installing a program at the worker's device. However, it is a strong assumption because the services need to persuade workers to install a program or install the software without the perception of workers.

Second, the services can handle every boosting task at the server. The services collect the tasks of workers and transmit the tasks to the target OSN when they wants. However, OSNs can recognize such activities because the same IP addresses are frequently used.

Third, the services can use bot accounts to secretly perform tasks. CrowdTarget may not work correctly if the services prepare an enough number of bot accounts to simulate the retweet time distribution of normal tweets. However, due to extra costs, we expect that the services would not take this approach.

7.2 Twitter Application

The crowdturfing services can use a large number of Twitter applications for evasion. By assigning different applications to different groups of workers, they can eliminate dominant applications. However, they cannot arbitrary create a large number of Twitter applications because Twitter restricts the number of application creation per day and per account. Furthermore, it is difficult to exactly control the ratio of the most dominant application, since workers can retweet any tweet at any time.

7.3 Unreachable Retweeters

To reduce the number of unreachable retweeters, the crowdturfing services would request crowdturfing workers to follow the posting user of a tweet they want to retweet. However, due to three important reasons, it is impractical. First, workers should receive future tweets of the posting user even if they do not want it. Second, increasing the number of followings can decrease the popularity of workers on Twitter, which is exactly opposite to their goal. Third, workers cannot follow the posting user when the number of their followers is small or when they recently follow many accounts [27].

7.4 Click Information

To manipulate the number of clicks, the crowdturfing services can request crowdturfing workers to click on a link in a tweet while retweeting it. This approach could evade CrowdTarget, but it has two problems. First, crowdturfing workers unwilling to click on such a link because it may be a malicious link (e.g., spam, phishing, and drive-by downloads). Second, we expect that the distributions of artificial clicks in terms of time, geographical location, user agents, and referrers differ from those of real clicks. Note that all links shared on Twitter are automatically shortened to `t.co` links [28]. This allows Twitter to obtain detailed click information of all links on Twitter. Thus, generating realistic click patterns by using crowdturfing workers would be a difficult task. Unfortunately, to the best of our knowledge, no crowdturfing service currently manipulates the number of click such that we cannot confirm our expectation. Therefore, in future, we will show how much effort is necessary to produce realistic click distributions.

8. RELATED WORK

In this section, we explain related studies of our work.

8.1 Detection of Crowdturfing Accounts

Malicious crowdsourcing has recently received considerable attention. Motoyama *et al.* [20] analyze various types of abuse tasks in Freelancer, one of the most popular crowdsourcing site. Wang *et al.* [33] collect data from crowdturfing sites based in China, Zhubajie and Sandaha, and analyze their structures, scale, and the amount of money involved in it.

Several researchers propose methods to detect crowdturfing aiming at OSNs. Lee *et al.* [18] and Wang *et al.* [32] aim

Table 2: Example of verified accounts that received retweets from accounts using automated applications

Verified accounts	Application name	Application homepage
PopWrapped	TweetAdder	`http://tweetadder.com`
m_bukairy	rtwity	`http://www.rtwity.com`
ODEONCinemas	Twitaculous	`http://twitaculous.com`
alweeamnews	twittretweet_EEE	`http://twittretweet.com`
CaesarsPalace	Social Rewards	`http://web.socialrewards.com`
Almatrafi	rettwwet_net	`http://rettwwet.net`
MohammadMamou	KLILK API RETWEET	`http://www.klilk.com`

to detect OSN accounts performing crowdturfing tasks on Twitter and Weibo, respectively. These studies use account-based features introduced in conventional spam detection studies, such as the ratio of tweets including links, the number of tweets per day, and the number of retweets per tweet. Lee *et al.* [19] detect malicious tasks targeting Twitter in Fiverr, one of the popular crowdsourcing site.

8.2 Detection of Malicious Accounts

There are a large number of studies of detecting malicious accounts in OSNs. We classify them into three types: account-based methods, graph-based methods, and behavior-based methods. First, account-based methods [12,17,23,34, 35] extract various features from user profiles and postings, and use them to build machine-learning classifiers. Second, graph-based methods [9, 10, 13, 30, 36, 37] detect malicious accounts by using the observation that malicious accounts usually have few connections with normal accounts. Third, recent researchers detect malicious accounts by monitoring their synchronized group activity. For example, COMPA [15] detects compromised accounts by catching similar changes of account behavior within a short time. Clickstream [31] classifies accounts based on the similarity of clickstream sequences. CopyCatch [8] and SynchroTrap [11] detect malicious accounts that have synchronized Facebook like patterns. CatchSync [16] uses synchronicity and normality of accounts to detect malicious accounts.

8.3 Detection of Black-market Accounts

Some researchers focus on black markets for OSNs. Stringhini *et al.* [24] analyze Twitter follower markets. They describe characteristics of Twitter follower markets and classify customers of the markets. Thomas *et al.* [25] investigate black-market accounts used for distributing Twitter spams. Cristofaro *et al.* [14] analyze Facebook like farms by deploying honeypot pages. Viswanath *et al.* [29] detect black-market Facebook accounts based on their like behaviors.

9. CONCLUSION

In this paper, we proposed a novel crowdturfing detection method using target objects of crowdturfing tasks, CrowdTarget. We observed that the manipulation patterns of the target objects maintained, regardless of what evasion techniques crowdturfing accounts used. Through the observation, we distinguished tweets that received retweets by crowdturfing sites from tweets that received retweets by normal Twitter users. Evaluation results showed that CrowdTarget could detect crowdturfing retweets on Twitter with TPR of 0.98 at FPR of 0.01.

Acknowledgments

We would like to appreciate our shepherd Guoliang Xue and anonymous reviewers for their invaluable comments and suggestions. This work was supported by ICT R&D program of MSIP/IITP. [14-824-09-013, Resilient Cyber-Physical Systems Research]

10. REFERENCES

[1] Addmefast. `http://addmefast.com/`.

[2] Embedded tweets. `https://dev.twitter.com/web/embedded-tweets/`.

[3] Klout. `https://klout.com/`.

[4] Retweets.pro. `http://retweets.pro/`.

[5] Socialshop. `http://socialshop.co/`.

[6] Traffup. `http://traffup.net/`.

[7] Twitter reaches spam lawsuit settlement with tweet adder. `http://marketingland.com/twitter-reaches-spam-lawsuit-settlement-with-tweet-adder-45890/`.

[8] A. Beutel, W. Xu, V. Guruswami, C. Palow, and C. Faloutsos. CopyCatch: Stopping group attacks by spotting lockstep behavior in social networks. In *International World Wide Web Conference (WWW)*, 2013.

[9] Y. Boshmaf, D. Logothetis, G. Siganos, J. Lería, J. Lorenzo, M. Ripeanu, and K. Beznosov. Íntegro: Leveraging victim prediction for robust fake account detection in OSNs. In *Network and Distributed System Security Symposium (NDSS)*, 2015.

[10] Q. Cao, M. Sirivianos, X. Yang, and T. Pregueiro. Aiding the detection of fake accounts in large scale social online services. In *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2012.

[11] Q. Cao, X. Yang, J. Yu, and C. Palow. Uncovering large groups of active malicious accounts in online social networks. In *ACM Conference on Computer and Communications Security (CCS)*, 2014.

[12] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia. Who is tweeting on Twitter: Human, bot, or cyborg? In *Annual Computer Security Applications Conference (ACSAC)*, 2010.

[13] G. Danezis and P. Mittal. SybilInfer: Detecting Sybil nodes using social networks. In *Network and Distributed System Security Symposium (NDSS)*, 2009.

[14] E. De Cristofaro, A. Friedman, G. Jourjon, M. A. Kaafar, and M. Z. Shafiq. Paying for likes?: Understanding Facebook like fraud using honeypots. In *Internet Measurement Conference (IMC)*, 2014.

[15] M. Egele, G. Stringhini, C. Kruegel, and G. Vigna. COMPA: Detecting compromised accounts on social networks. In *Network and Distributed System Security Symposium (NDSS)*, 2013.

[16] M. Jiang, P. Cui, A. Beutel, C. Faloutsos, and S. Yang. CatchSync: Catching synchronized behavior in large directed graphs. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2014.

[17] K. Lee, J. Caverlee, and S. Webb. Uncovering social spammers: Social honeypots + machine learning. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2010.

[18] K. Lee, P. Tamilarasan, and J. Caverlee. Crowdturfers, campaigns, and social media: Tracking and revealing crowdsourced manipulation of social media. In *International AAAI Conference on Web and Social Media (ICWSM)*, 2013.

[19] K. Lee, S. Webb, and H. Ge. The dark side of micro-task marketplaces: Characterizing Fiverr and automatically detecting crowdturfing. In *International AAAI Conference on Web and Social Media (ICWSM)*, 2014.

[20] M. Motoyama, D. McCoy, K. Levchenko, S. Savage, and G. M. Voelker. Dirty jobs: The role of freelance labor in web service abuse. In *USENIX Security Symposium*, 2011.

[21] Scikit-learn. `https://http://scikit-learn.org`.

[22] J. Song, S. Lee, and J. Kim. I know the shortened URLs you clicked on Twitter: Inference attack using public click analytics and Twitter metadata. In *International World Wide Web Conference (WWW)*, 2013.

[23] G. Stringhini, C. Kruegel, and G. Vigna. Detecting spammers on social networks. In *Annual Computer Security Applications Conference (ACSAC)*, 2010.

[24] G. Stringhini, G. Wang, M. Egele, C. Kruegel, G. Vigna, H. Zheng, and B. Y. Zhao. Follow the green: growth and dynamics in Twitter follower markets. In *Internet Measurement Conference (IMC)*, 2013.

[25] K. Thomas, D. McCoy, C. Grier, A. Kolcz, and V. Paxson. Trafficking fraudulent accounts: The role of the underground market in Twitter spam and abuse. In *USENIX Security Symposium*, 2013.

[26] Twitter. State of Twitter spam. `https://blog.twitter.com/2010/state-twitter-spam`.

[27] Twitter Blogs. Following rules and best practices. `https://support.twitter.com/entries/68916-following-rules-and-best-practices`.

[28] Twitter Blogs. Next steps with the t.co link wrapper, 2011. `https://blog.twitter.com/2011/next-steps-with-the-tco-link-wrapper`.

[29] B. Viswanath, M. A. Bashir, M. Crovella, S. Guha, K. P. Gummadi, B. Krishnamurthy, and A. Mislove. Towards detecting anomalous user behavior in online social networks. In *USENIX Security Symposium*, 2014.

[30] B. Viswanath, A. Post, K. P. Gummadi, and A. Mislove. An analysis of social network-based Sybil defenses. In *ACM SIGCOMM*, 2010.

[31] G. Wang, T. Konolige, C. Wilson, X. Wang, H. Zheng, and B. Y. Zhao. You are how you click: Clickstream analysis for Sybil detection. In *USENIX Security Symposium*, 2013.

[32] G. Wang, T. Wang, H. Zheng, and B. Y. Zhao. Man vs. machine: Practical adversarial detection of malicious crowdsourcing workers. In *USENIX Security Symposium*, 2014.

[33] G. Wang, C. Wilson, X. Zhao, Y. Zhu, M. Mohanlal, H. Zheng, and B. Y. Zhao. Serf and turf: Crowdturfing for fun and profit. In *International World Wide Web Conference (WWW)*, 2012.

[34] C. Yang, R. C. Harkreader, and G. Gu. Die free or live hard? empirical evaluation and new design for fighting evolving Twitter spammers. In *Recent Advances in Intrusion Detection*, pages 318–337. Springer, 2011.

[35] Z. Yang, C. Wilson, X. Wang, T. Gao, B. Y. Zhao, and Y. Dai. Uncovering social network Sybils in the wild. In *Internet Measurement Conference (IMC)*, 2011.

[36] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao. SybilLimit: A near-optimal social network defense against Sybil attacks. In *IEEE Symposium on Security and Privacy (Oakland)*, 2008.

[37] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. SybilGuard: Defending against Sybil attacks via social networks. In *ACM SIGCOMM*, 2006.