Noname manuscript No.
(will be inserted by the editor)

Analysis of and Defense against Crowd-Retweeting based Spam in Social Networks

Bo Liu · Zeyang Ni · Junzhou Luo · Jiuxin Cao · Xudong Ni · Benyuan Liu · Xinwen Fu

Received: date / Accepted: date

Abstract Social networking websites with microblogging functionality, such as Twitter or Sina Weibo, have emerged as popular platforms for discovering real-time information on the web. Like most Internet services, these websites have become the targets of spam campaigns, which contaminate web contents and damage user experiences. Spam campaigns have become a great threat to social network services. In this paper, we investigate crowd-retweeting spam in Sina Weibo, the counterpart of Twitter in China. We carefully analyze the characteristics of crowd-retweeting spammers in terms of their profile features, social relationships and retweeting behaviors. We find that although these spammers are likely to connect more closely than legitimate users, the underlying social connections of crowd-retweeting campaigns are different from those of other existing spam campaigns because of the unique features of retweets that are spread in a cascade. Based on these findings, we propose retweeting-aware link-based ranking algorithms to infer more suspicious accounts by using identified spammers as seeds. Our evaluation results show that our algorithms are more effective than other link-based strategies.

Keywords Social Network · Crowd-Retweeting · Spamming and Microblogging

1 Introduction

Microblogging social networks such as Twitter [32] and Sina Weibo [35] have emerged as popular platforms for discovering real-time information on the Inter-

Bo Liu · Zeyang Ni · Junzhou Luo · Jiuxin Cao · Xudong Ni
School of Computer Science and Engineering, Southeast Univerisity, NanJing, China 211189
E-mail: { bliu, nizy, jluo, jx.cao, xd_ni}@seu.edu.cn

Benyuan Liu
Department of Computer Science, University of Massachusetts Lowell, MA, USA
E-mail: :bliu@cs.uml.edu

Xinwen Fu
Department of Computer Science, University of Central Florida, FL, USA
E-mail: :xinwenfu@cs.ucf.edu

net, such as news, events, and opinions. It has been reported that Twitter has over 328 million monthly active users as of 2017 and generates over 500 million tweets on a daily basis [28], while Sina Weibo as China's most popular microblog website, has 313 million monthly active users as of February 2017; 92 million messages were being posted each day as of that time [7].

Due to their popularity and the ease with which messages can be spread, microblogging social networks also have become targets of spam and phishing scams [14]. Spammers use microblogging systems to advertise sales, to phish, to disseminate pornography, and to spread viruses. The spam messages not only pollute real-time search and statistical mining results but also consume user and system resources. Spam detection in Twitter, particularly, has already caught the attention of researchers [1, 18, 39]. These researchers found that spam detection in social networks is different from traditional spam detection, which often assumes that spam messages are spread by software bots. The reason is that spamming strategies in social networks are changing.

The spamming strategy has been evolving; for example, a crowdsourcing system on the Internet can be used to hire large numbers of real individual users to perform a task. It is not surprising that spam campaigns have already adopted crowdsourcing systems to spread spam in the social networks and to try to avoid traditional spam detection [5, 24, 34]. It has been reported that the US military has developed specific software to speed up the distribution of pro-American propaganda in social media [10]. In China, the "Internet Water Army" is a group of paid writers posting on social media to advertise or to manipulate the public opinion [5, 36]. Such spam campaigns tend to select users with a high indegree in the social graph to propagate spams because the more followers a user has, the more likely his or her retweets spread widely and quickly.

Spam campaigns in Twitter-like social networks usually hire a large number of real labor workers to *retweet* unsolicited ads or controversial topics with comments to increase the audience size and influence. This crowd-retweeting spamming poses a serious threat to microblog-based social networks for several reasons. Firstly, it takes advantage of a social networks's inherent information cascade effect, which spreads spam messages quickly and reaches a large audience. Secondly, employed spammers retweet the target messages manually to get paid, and can evade traditional spammer detection strategies. Lastly, these real spammers also use various tactics to make their accounts look attractive and normal by periodically updating tweets and following other users to avoid being labeled as Zombie accounts.

In this paper, we investigate crowd-retweeting spamming in the Sina Weibo (Weibo) social network. We crawl a large dataset of Weibo accounts using data from real paid posters as our seeds, which are obtained from two public crowdsourcing service providers - Zhubajie (ZBJ) [44] and Sandaha (SDH) [25]. There are two types of crowd-retweeting spammers: a *spam initiator* pays *spam workers* to retweet target spam messages. We carefully study and analyze profile features, social relationships and retweeting behavior of both spam initiators and spam workers. We propose two inference algorithms utilizing the link structure and retweeting characteristics to find more suspicious spam initiators and workers. The experiment results show that our methods perform well to identify potential spammers.

The main contributions of this paper can be summarized as follows. Firstly, we collect reliable data of spammers from real crowdsourcing websites ZBJ and SDH,

which are credible foundations for the analysis of the spammer social structure. Secondly, we analyze spammer characteristics and find the unique features of spam initiators and workers. We find that spam initiators tend to be connected nonreciprocally and act like normal users. In contrast, spam workers are more closely connected among themselves with strong social relationships and their retweeting behavior is different from normal users. Thirdly, As an extended and improved version of our previous works [20], we dive deeper into the characteristics of spammers and introduce more efficient inferring algorithms based on both link structure and retweeting features to find more suspicious spam workers and spam initiators. Especially, we introduce a new way to measure those workers holding few following relationship with workers in the seed set, which were hardly detected in previous work.

The rest of this paper is organized as follows. We present related work in Section 2. Section 3 introduces how we collected real-world data. Section 4 provides our detailed analysis of spammers' characteristics. Section 5 presents our algorithms to detect such spammers. Evaluation results are presented in Section 6. Section 7 concludes this paper.

2 Related Work

In this section, we briefly discuss related literature on the crowd-retweeting-based spam by organizing it into three groups: crowdsourcing systems, link structure-based analysis, and spam detection.

2.1 Crowdsourcing Systems

Crowdsourcing systems have been widely studied on the Internet. Amazon's Mechanical Turk [31] is best known crowdsourcing platform and has been scrutinized by many researchers [15]. Other systems like Freelancer [11], Microworkers [23] have also been studied [22]. The abuse of crowdsourcing systems has been discussed in recent research works. For example, Motoyama et al. [24] characterized the behaviors of abuse-related labors on Freelance.com, such as account registration, ad posting and social network link farming. Chen et al. [5] analyzed the hidden paid posters and proposed an SVM-based detection scheme using posters' behavior features and postings' semantic similarity. Due to the limitations of their dataset, the authors were not able to investigate similar malicious users in social networks. Wang et al. [34] confirmed the existence of the malicious usage of crowdsourcing systems by analyzing two crowdturfing websites ZBJ and SDH. They compared five types of campaigns and checked profiles of suspicious accounts as well as the effect of crowdturfing on information dissemination in Weibo. In [40], the authors analyzed the crowdturfing phenomenon in Sina Weibo and proposed an LDA-based method to investigate the users who often posted political contents. However, their analysis ignored the underlying structure of the social network.

2.2 Link Structure-based Analysis

Using link structure to design security schemes involves web link farming and Sybil attack defense. Link farming is a form of web spam targeting link-based ranking algorithms. Gyongyi et al. presented TrustRank to propagate trust based on websites' links [13]. The main idea is that good pages are likely to point to other good sites. Becchetti et al. proposed link-based techniques for detection of web spam using probabilistic counting [2]. However, the link structure of people's relationships is different from that of websites.

Other researchers explore the properties of social networks to defend against Sybil attacks. SybilGuard [43] was the first work exploiting the link structure to contain Sybil attacks's influence. Similar strategies like SybilLimit [42], SybilInfer [8], and SumUp [30] have also been proposed. Viswanath et al. [33] summarized these schemes and found they are vulnerable when non-Sybil nodes form strong communities. Sybil defense systems can work well only when non-Sybil nodes form a single community that is distinguishable from the group of Sybil nodes. Yang et al. found that Sybils in reality do not form tight-knit communities [41]. Xue et al. proposed a scheme using the friend invitation graph to defend against social network Sybils [37]. However, crowd-retweeting based spammers in Twitter-like social networks are different from traditional Sybils and thus need more research to be fully understood.

2.3 Spam Detection in Social Networks

Social network spam detection methods can be roughly classified into two categories: machine learning and link based inference. Lee et al. [18] proposed a classification approach by deploying social honeypots. Thomas et al. [29] made an in-depth study of spammers' behavior in Twitter to understand the spam ecosystem. A spam classification approach is proposed based on user profile features [3] . In [19], the authors studied features including profile information, comment behavior, IP address as well as geographical information and used SVM to detect spammers. Chen et al. [4] analyzed the click rate of spam messages in Twitter, and proposed a spammer detecting approach based on existing malicious link detection technologies.

Recently, researchers have begun to pay attention to spammer communities' social structures. Yang et al. [38] found that malicious users are socially connected and proposed an inference algorithm based on identified malicious accounts. They used malicious URLs to define spammers not the same as the crowdsourcing spammers studied in this paper. Ghosh et al. [12] investigated the link farming in Twitter and presented a scheme using links to reduce the rank of malicious users. However, they considered only following links to reduce the users' rank. It has been already found [17] that ranking by following relationships and retweeting can get different results. Fayazi et al. [9] considered users' comment behavior and introduced a Markov random field-based method to infer spammers. Nevertheless, their proposed model focuses mainly on electronic business websites, which are quite different from microblog websites studied in this paper. As noted above, for various reasons, the methods discussed above are not sufficient for detecting crowd-reweeting spammers.

In this paper, our data collection strategy is very different from those used in the works above. We use data from real paid posters as our seed set to crawl more data. We find that although spammers are connected closely, their relationships are often undirectional. Spammers do not just gain influence by *following*; they also spread spams by retweeting. This makes the strategy of using only the *following* link to detect spammers insufficient.

3 Data collection in Weibo

In this section, we will first demonstrate how a malicious crowdsourcing system can be built upon social networks for spamming. We will then introduce our methodology of studying how malicious crowdsourcing system users distribute spam and our method of data collection. We present the terminology used in this paper at last.

3.1 Combining Crowdsourcing Systems and Social Networks

As shown in Fig. 1, a crowd-retweeting spam campaign has three key actors: *Initiator*. An initiator is an individual or a company who initiates a spam campaign and pays the cost. For example, an initiator can initiate a campaign by posting on Weibo. An initiator can also intentionally send tasks to an agent, for example, posting a task at an agent's website. *Agent*: Agents are intermediaries who take charge of finding tasks, managing and distributing funds to workers to accomplish the goals. Agents can be in different forms such as websites and forums. For example, ZBJ and SDH are web based services acting as agents. An initiator can hire spam workers from ZBJ or SDH, asking these workers to distribute his or her post on Weibo. *Worker*: Workers are paid posters who perform tasks assigned by agents. In Weibo, spam workers retweet or comment on tweets of an initiator to attract other normal Weibo users and will get paid for the work.

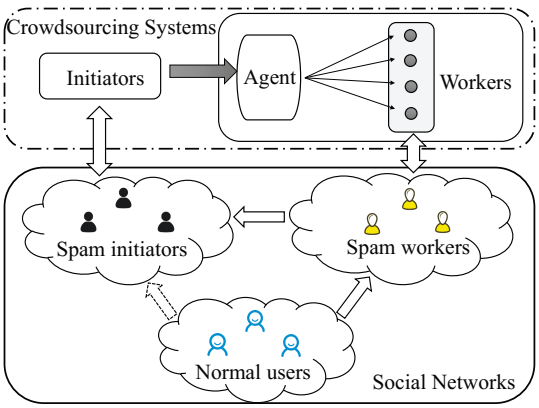


Fig. 1 Mapping of Crowdsourcing Systems and Social Networks.

A typical crowd-retweeting spam campaign in Weibo may work as follows:

1. An initiator has a spam task and asks the agent to carry it out.
2. The agent publishes the task and distributes it to spam workers.
3. After completing the tasks, workers report to the agent, who passes the report back to the initiator.
4. If the initiator is satisfied with the result, he will inform the agent to make payments to the workers.

3.2 Methodology

To study how crowd-retweeting based spamming works in the Weibo network, we need (i) a large number of Weibo spammers' identities that come from real crowdsourcing systems and (ii) a large dataset of Weibo user accounts including normal users and spammers. So we collect our dataset in two steps. First, we choose two public available websites ZBJ and SDH as our crowdsourcing sources, which are also used in related work [34] although the authors do not explore the social relationship link structure and retweeting behavior. We developed a web crawler based on Crawler4j [6] and crawled ZBJ and SDH webpages that contain Weibo's spamming tasks. These spammer data from ZBJ and SDH can be used for exploring spammer characteristics and also work as seeds to find other spammers. Then, with the help of Sina Weibo API we crawled these spammers' profiles, follower lists, followee lists, tweets and retweets. The crawling tasks above were completed by 2012. In order to avoid the bias caused by single dataset, a new dataset with different time span is needed to cooperatively analyze the characteristics of crowd-retweeting spamming. We developed another web crawler based on Scrapy [26] and crawled more crowd-retweeting spam campaign tasks from ZBJ and SDH and spammer data from Weibo from Jan. 2015 to Dec. 2016. Therefore, we have two datasets from different time periods.

To extract a spammer's identity, we need firstly define the spam. The definition of spam in social network is slightly different from previous ones, which define tweets containing malicious URLs as a spam. According to Twitter, spammers are those users who post harmful links, abuse the @reply or @mention function to post unwanted messages to users, and post repeatedly about trending topics to grab attention. It can be observed that the crowd-retweeting based spamming in social network matches the definition from Twitter.

Since spammers in our data take two roles: initiator and worker, we need to identify initiator and worker accounts to make an in-depth analysis. Here is how agents like SDH or ZBJ post tasks. The agent collects a list of tasks and publish them in a webpage. Clicking on a task in the list, a user is directed to a topic thread and the thread initiator is the spam initiator, who specifies the particular task with his or her Sina Weibo identity in his or her post. Workers reply to the post from the thread initiator, posting the snapshots of retweets to get paid. Our script can analyze the crawled webpage structure and extract the Weibo user ids(uids) of those initiators. By crawling each thread, we can also obtain Weibo uids of those workers for each task.

The goal of this paper is to design algorithms and identify *unknown* spam initiators and workers based on tweets on Weibo. Note: spammers identified through ZBJ and SDH may be only a small fraction of spammers active on Weibo. This

is why we need automatic approaches to identify suspicious spammers. We shall crawl data on Weibo in order to understand the behavior of spammers and normal users so that we can differentiate spammers from normal users. Therefore, spammer uids obtained from ZBJ and SDH and randomly chosen normal user uids are used as seeds for our crawlers to obtain a large dataset, which includes uids, profiles, follower lists, followee lists, tweets and retweets.

The architecture of crawler based on Crawler4j is shown in Fig. 2. The crawler was developed based on Sina Weibo Open SDK [27] in 2012. Weibo's SDK used the OAuth 2.0 protocol to delegate authentication and it also had a API rate limit. For example, for an authorized application, it had a total rate limit of 1,000 calls per hour, with each test user account having a rate limit of 150 calls per hour. To maximize the API call hits, we applied for 30 applications and 7 users, obtaining about 30,000 API hits per hour for our crawler. Then we designed a Breadth-First-Search (BFS) scheme to keep our crawler collecting more nodes. We chose the users' followees as our crawling candidates, since some nodes had a millions of followers, which would soon consume the API hits. For potential spammers, we collected the information of 14,443 users out of 14,536 while the rest had already been banned by Weibo official administrator. Finally, we got 3,658 spam initiators and 11,045 spam workers. We also collected a dataset sample that contained 193,591 users and 10,785,921 tweets from a randomly generated seed set.

The Scrapy based crawler used a similar crawling logic. We crawled more crowd-retweeting spam tasks from ZBJ and SDH. After removing those banned accounts, we got 892 new spam workers and 706 new spam initiators. Furthermore, information of 38,948 Weibo users were collected from another randomly generated seed set. In total, we crawled 40,546 users along with 8,321,270 tweets with Scrapy.

Specifically, the dataset collected by 2012 is named as Dataset A and Dataset B for the one collected from 2015 to 2016. And we randomly picked up three small normal user samples from both two datasets for comparison with spammers. Each normal user random sample had around 1,500 Weibo accounts.

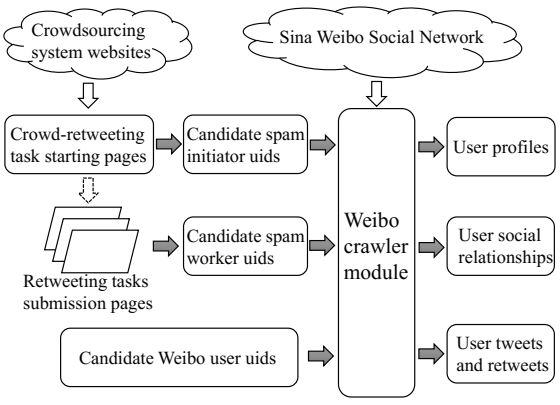


Fig. 2 Weibo Crowd-Retweeting Crawler Architecture.

3.3 Terminology

Fig. 3 introduces the terminology we use in the rest of this paper. If node (user) B follows node A, we refer to B as A's follower and A as B's followee. We can represent the relationship in the social graph with a directed edge from B to A, which also increase A's indegree and B's outdegree by 1. We also diffferentiate the tweets and retweets. Each node C has a list of tweets called timeline. If a tweet is posted by the node itself, we call it an original tweet. If other nodes retweet or comment on this tweet, the original tweet along with its retweets will form a retweet tree [17]. If a node retweets a message from its followee node, we call it one hop forward, which is quite common in a crowd-retweeting propagation procedure. As shown in Fig. 3, the dashed arrow line from F to C is one hop forward retweeting path while the line from D to C is not. A typical crowd-retweeting based spamming includes a spam initiator node who posts an original tweet and spammer worker nodes who massively retweet it.

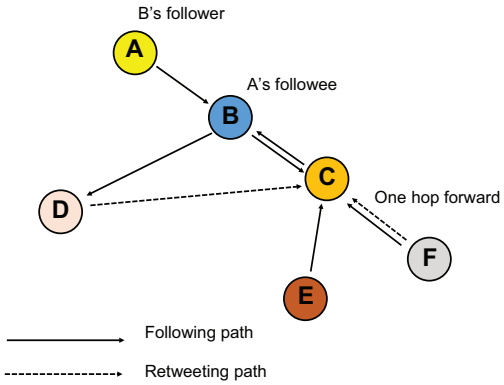


Fig. 3 Terminology of Crowdsourcing based Social Network.

4 Analysis of Crowd-retweeting Spamming

In this section, we perform statistical analysis of spammers in a crowd-retweeting spamming campaign. Our analysis mainly uses profile features, social structure, and retweeting behavior.

4.1 Profile Characteristics

We start our analysis by obtaining profiles of Weibo accounts. The profiles can provide features such as the number of followers, followees and tweets. These statistics often reflect the reputation of Weibo accounts in a social network. They are often used as features by spam detection algorithms [1,3]. To verify whether these features are still useful, we compare the profiles of spam workers and initiators

as well as randomly selected normal users in three normal user samples. The statistical analysis is carried out on both two datasets to study the stability of the profile characteristics we analyzed. Specifically, Fig. 4 (a) (c) (e) show the statistics on Dataset A while Fig. 4 (b) (d) (f) on Dataset B. Compared with normal users, spammers have more followees and followers. For example, Fig. 4 (a) is the empirical cumulative distribution function (ECDF) of users' followee numbers. In Fig. 4 (a), 88% of normal users follow no more than 500 users, while only 45% of spammers follow fewer than 500 users. The curve has a sharp turn around 2,000 followees for spammers because Weibo has a upper bound of the number of a user's followees. VIP users can follow more users by paying an annual fee. Similar observation can be concluded from Fig. 4 (b), which shows the following pattern on Dataset B.

From Fig. 4 (c) on Dataset A and Fig. 4 (d) on Dataset B, despite the small difference of analysis results between two datasets, spam workers have more followers than normal users. This is caused by the spamming mechanism. For a typical spam task, the spam initiator announces a total reward and each participated worker can only be paid one time. The payment is in proportion to the number of a spammer worker's followers. For instance, if a worker has 50 to 100 followers, he can earn 0.05 to 0.1 CNY per post while a worker with 100 to 500 followers can get 0.1 to 0.5 CNY. For those who have twenty millions of followers, they can earn about 20,000 CNY for each retweet. This is why spammers need so many followers.

According to Fig. 4 (a) (b) (c) (d), although there is a certain degree of difference on trajectory of curves between initiators and normal users, it can be observed that the following pattern of initiators is more similar to the following pattern of normal users compared to that of workers. This is because spam initiators are not responsible for spreading spam.

An interesting finding is that the distribution of the number of spammers' tweets is similar to the distribution of the number of normal users' tweets as shown in Fig. 4 (e) on Dataset A and (f) on Dataset B. There is no big difference between the number of spammers' tweets and the number of normal users' tweets; and even the number of spammers' tweets are a little bit less than the number of normal users' tweets. This is because spam workers prefer retweeting initiators' tweets which are profitable instead of tweeting many normal posts; and also part of normal users have many tweets, for example, VIP users update their tweets frequently to attract more followers. And imitate the tweeting behavior of normal users can alse help to defeat Weibo's fraud detection mechanism.

In summary, we find the number of followees and followers can differentiate spammers from normal users to some extent, but the number of posted tweets is less useful while it has been utilized in previous spam detections.

4.2 Social Relationship Characteristics

In this section, we study the social structure, relationship characteristics and tweeting behavior of spammers in our dataset. Intuitively, spammers are supposed to connect closely and have some special relationship characteristics [38]. We perform various statistical analysis to identify social relationship features of spammers in Weibo and report our findings below.

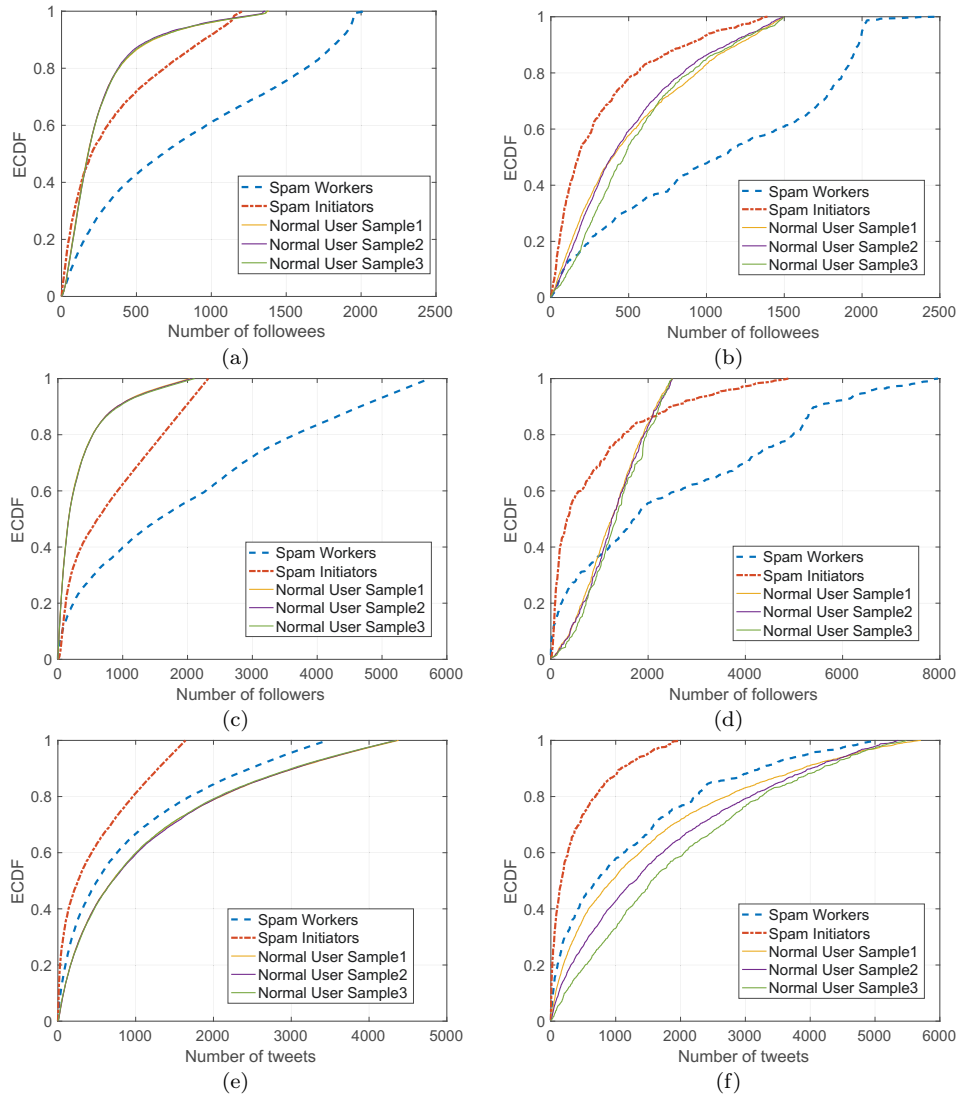


Fig. 4 (a) ECDF of the number of followees of spam workers, spam initiators and three normal user samples on Dataset A. (b) ECDF of the number of followees of spam workers, spam initiators and three normal user samples on Dataset B. (c) ECDF of the number of followers of spam workers, spam initiators and three normal user samples on Dataset A. (d) ECDF of the number of followers of spam workers, spam initiators and three normal user samples on Dataset B. (e) ECDF of the number of tweets of spam workers, spam initiators and three normal user samples on Dataset A. (f) ECDF of the number of tweets of spam workers, spam initiators and three normal user samples on Dataset B.

4.2.1 An Overview of Sample Social Graph

If we view each Weibo account as a node v and each following relationship as directed edge e, we can then model the social network as a directed graph $G =$

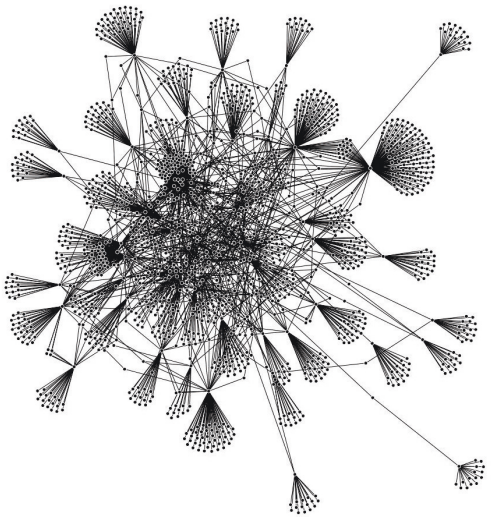


Fig. 5 Spammers Social Graph Containing 2,500 Edges.

Table 1 Summary of Dataset's Metrics

Dataset	#Nodes	#Edges	#INodes	#WCC	LCS
Spammers	14,443	461,415	421	1	14,022
Sample 1	14,987	7,337	9,841	456	4,068
Sample 2	14,990	7,274	10,111	508	3,717
Sample 3	14,983	8,480	9,344	388	4,740
Sample 4	14,988	7,215	10,149	455	3,802
Sample 5	14,993	8,865	9,299	401	4,757
Spam Initiators	3,567	20,768	331	3	3,232
Spam Workers	10,876	167,198	949	30	8,759

(V, E). In this paper, we only consider the inner social relationships. That is, we collect only the edges whose vertices are in our dataset. Table 1 shows the metrics of our social graphs. The 4th column (#INodes) is the number of isolated nodes. The 5th column (#WCC) is the number of weakly connected components and the 6th column (LCS) is the largest component's size. A sample social graph that contains 2,500 following edges from the spammer dataset is shown in Fig. 5. We can see that the major portion of the spammers form a closely connected community. Some spammers in the outer layer are sparsely connected. That is, a small portion of spammers do not have much following relationships with other spammers.

4.2.2 Revealing Relationship Characteristics

After calculating various graph metrics of the relationship graph of our samples, we have the following observations:

Finding 1: Spammers are closely connected compared with normal users. To quantitatively validate this finding, we use two graph metrics: graph density and reciprocity. Graph density is the ratio of the number of edges over the number of possible edges. A higher value implies that the graph is denser. We find that the spammer graph's density is 2.21×10^{-3} while the graph density of the five normal

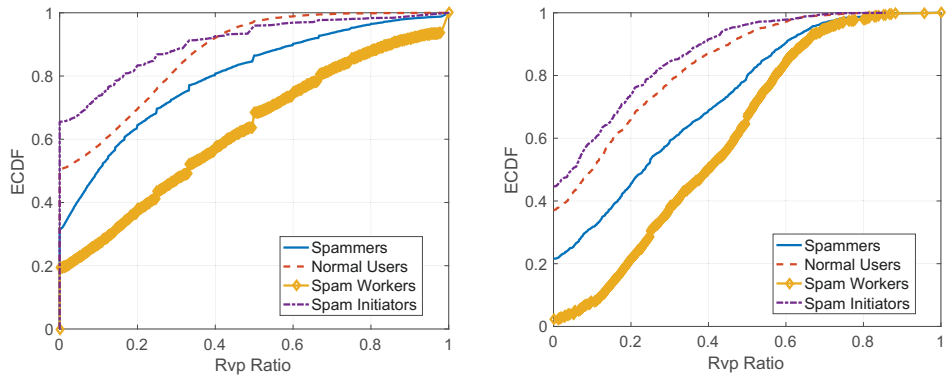


Fig. 6 ECDF of the RVP Ratio for (a) Spammers vs Resampled Normal Users vs Spam Workers vs Spam Initiators on Dataset A. (c) Spammers vs Resampled Normal Users vs Spam Workers vs Spam Initiators on Dataset B.

user samples is 3.27×10^{-5} (Sample 1), 3.24×10^{-5}(Sample 2), 3.78×10^{-5}(Sample 3), 3.21×10^{-5}(Sample 4) and 3.94×10^{-5}(Sample 5) respectively. We also find that the graph density of spam initiators is 1.63×10^{-3} and graph density of spam workers is 1.41×10^{-3}. This shows that the spammers have closer relationships than normal users.

Reciprocity defines the proportion of mutual connections in a directed graph. It can be calculated as $r=\frac{L^{\leftrightarrow}}{L}$, where $L^{\leftrightarrow}$ is the number of reciprocal links and L is the total number of links. Fig 6 is the ECDF of reciprocated vertex pair (*RVP*) ratio of four kinds of user groups on Ddataset A as well as Dataset B. Both two figures show that the RVP ratio of spammers (including both workers and initiators) is slightly lower than normal users'. It means that spammers connect closely to form a community in the social networks. The reason is a crowd-retweeting based spamming task requires spammer workers to keep following spam initiators. Spammer workers often register with different agents and submit a large number of tasks so that their outdegree increases. The effect leads to the closeness between spammers.

Finding 2: Within the spammer social community, spam workers are more likely to follow each other to form a small world while spam initiators tend to be connected nonreciprocally and their behavior is similar to normal Weibo users. This finding is derived in the following way. We first divide the spammers into subgraphs. Then we compute and analyze the RVP ratio of spam workers and spam initiators. As Fig. 6 (a) shows, for spam workers, we find 70% of the RVP ratios are greater than 0.2, quite different from normal users. But for the spam initiators, only 20% of the RVP ratios are greater than 0.2, similar to the RVP ratio of normal users. The possible reason is that spam workers follow each other to obtain more job opportunities. By following each other, they can discover tasks submitted by other workers and get involved in those tasks quickly to make more money. However, spam initiators have no intention to follow spam workers since the spam workers will report to the initiators at ZBJ or SDH to get paid. Similar observations can be found on Dataset B in Fig. 6 (b).

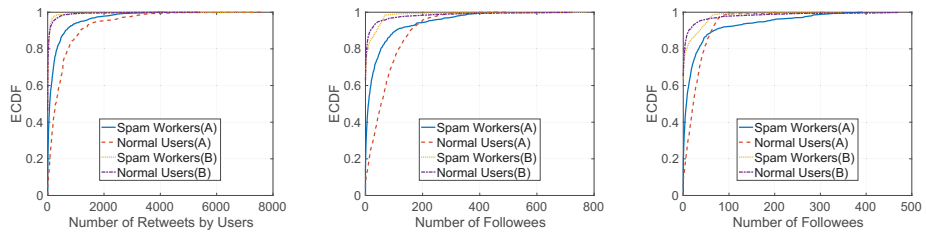


Fig. 7 (a) ECDF of the number of retweets by users (b) ECDF of the number of followees whose posts have been retweeted by spam workers and normal users (c) ECDF of the number of followees whose posts are one-hop retweeted by spam workers and normal users.

4.3 Retweeting Characteristics

Spam workers spread spam messages by retweeting or commenting so that these messages can be instantly updated in timelines of their followers. Intuitively, spam workers are more likely to retweet than other users. To verify this claim, we randomly choose one normal user sample and one spammer work sample of size 1000. Fig. 7 (a) gives the ECDF of the number of retweets by users. It can be observed that 90% spam workers, compared with 70% of normal users, retweet less than 500 times. This is counterintuitive since intuitively spam workers are more likely to retweet in order to make money. The statistics on Dataset B indicates the decrement of the number of retweets. It is due to the shorter collecting time span. Despite the difference in quantity, the findings on the two datasets are consistent.

Further analysis is performed to find the reason why spam workers retweet less and have observed unique features of spammers and we get a new observation:

Finding 3: In contrast to normal users, spam workers are more inclined to retweet the tweets of their followees, especially for one-hop retweeting. There are two types of retweeting by a user: retweeting posts created by accounts in the followee list of the user and retweeting posts created by other accounts. Fig. 7 (b) is the ECDF of the number of followees whose posts have been retweeted by spam workers and normal users. We find the two curves intersect at about 200. After the intersection, the ECDF for the spammer workers increases slowly. It implies that there are more spam workers who follow a large number of accounts and also retweet posts created by their followees than normal users. We go further to analyze the one-hop retweeting behavior. Fig. 7 (c) gives the ECDF of the number of followees whose posts are one-hop retweeted by spam workers and normal users. It can be observed that only 10% of spammers and normal users retweet posts of more than 50 followees. However, it is less likely that retweeted posts from spam workers are retweeted again by the followers of the spam workers.

4.4 Finding Discussions

Three key findings are observed after the analysis of crowd-retweeting spamming. And how these findings contribute to the spammer detection algorithms will be briefly discussed in this section. Separately, the three findings are:

Finding 1: Spammers are closely connected compared with normal users.

Finding 2: Within the spammer social community, spam workers are more likely to follow each other to form a small world while spam initiators tend to be connected nonreciprocally and their behavior is similar to normal Weibo users.

Finding 3: In contrast to normal users, spam workers are more inclined to retweet the tweets of their followees, especially for one-hop retweeting.

According to Finding 3, one-hop retweeting takes a large proportion in spam workers' retweeting behavior when they forward tweets from initiators. So we treat the relationship of one-hop retweeting the most important factor when inferring initiators, especially with a seed set of spam workers. In the detection of spam workers, we mainly take use of the following topology on the basis of Finding 1 and Finding 2. From Finding 3 it is known that spam workers usually share similar retweeting behavior, which is also taken into consideration in the spam worker detection algorithm.

5 Algorithms of Detecting Spammers

Considering the huge number of Weibo accounts, it is impractical to manually and deeply analyze every account to check whether it is a spammer account. Our basic idea of finding spammers is to design algorithms and search for suspicious accounts from a set of seed spammers, whose identities are known. Recall that there are two types of spammers: initiator and worker, in a crowd-retweeting spamming campaign. We design two algorithms to infer spam initiators and workers respectively while the seeds are spam workers.

5.1 Inferring Spam Initiator

Our Spam Initiator Inference Algorithm (SIIA) is a variant of the HITS [16] algorithm while considering the retweeting relationships given in Section 4 in order to find the spam initiators. In Section 4, we find that a worker is more likely to become the follower of an initiator than other social network users (*Finding 1*) and the worker inclines to forward or comment on an initiator's tweets than other followees of the worker (*Finding 3*). Therefore, an initiator can be viewed as an authority node with many incoming links while spam workers are hub nodes with many outgoing links. However, using the following relationships alone could lead to the tight-knit effect since spammers also massively follow other Weibo accounts, such as celebrities, making these nodes have a high authority score. This may incur a large false positive rate for the inference algorithm. So we consider one hop forward retweeting relationships given in Section 4. The reason is spam workers can only get paid by following initiators and forwarding their original tweets, making these following links more important than others. Instead of using the social graph G constructed by only the following relationships [1, 17, 38], we construct a subgraph G' of G and one hop forward retweeting list as the input of our algorithm. In Fig. 8, the solid arrow line represents the normal following relationship while the retweeting relationship is represented by the dashed arrow line. HITs uses the *following* edges to calculate the hub and authority score. Our SIIA algorithm only uses the nodes with one hop retweeting relationships. Therefore, we obtain a new graph with five nodes $\{A, B, C, D, E\}$. Authority and hub scores are calculated by

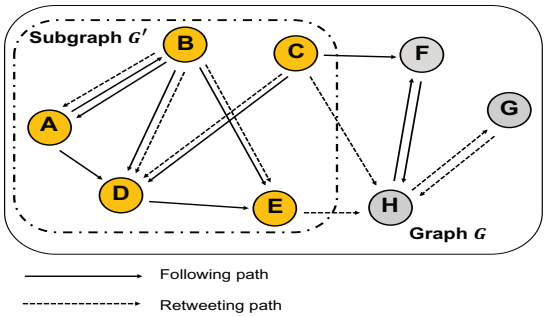


Fig. 8 Constructing Subgraph G' out of Graph G.

Algorithm 1 Spam Initiator Inferring

Input: social graph, $G = (V, E)$; set of known spammers, S; user's one hop retweet list, L
Output: Initiator Relevance score: IR
$G' = \emptyset$
for each user u in set S
  add u as a vertex to G'
  for each node v in u_i's one hop retweet list L
    if $v \in G$ **and** $v \notin G'$
      add v and e_{ij} to G'
    end if
  end for
end for
/*calculate authority score a and hub score h for G' */
/*initialize score vector d_a and d_h for all nodes n in G'*/
$d_a(n) = 1$, $d_h(n) = 1$
/*nf denotes the followees of node n, nfr denotes the followers of node n*/
while d_a or d_h not converged **do**
  for all nodes n in G' **do**
    $d_h(n) = \sum_{nf \in followees(n)} d_a(nf)$
    $d_a(n) = \sum_{nfr \in followers(n)} d_h(nfr)$
  end for
  Normalize d_a and d_h
end while
return $IR = d_a$

a series of iterations, with each iteration performing both authority update and hub update.

Algorithm 1 is the sketch of SIIA. In our algorithm, we select a set of identified spam workers as our seed nodes and derive a list of spam initiators ranked by their authority scores, denoted as Initiator Relevance (IR) scores in our context. We first construct a new graph G' according to the seed node's one hop retweeting list and following links and then calculate authority scores of all the nodes. A higher authority score implies the account is followed and retweeted by more users. And this corresponds to the behavior of a spam initiator. We can rank all suspicious initiator nodes and select top K accounts with a high authority score as spam initiators.

5.2 Inferring Spam Worker

Since spam workers are more closely connected than spam initiators, our basic idea is to infer spam workers based on a set of identified spam workers. We introduce the Worker Relevance(WR) score to measure an account's relevance with spam workers. To assign a WR score for each account, we design a Spam Worker Inference Algorithm (SWIA), which propagates the WR score from a seed set of real spammer accounts to their followers. An account with a larger WR score has a higher probability to be a spam worker and the account will have a higher rank in our spammer list.

Our algorithm is based on the findings in Section 4: (1) Spam Workers tend to be socially connected (*Finding 1* and *Finding 2*); (2) Spam Workers usually share similar retweeting behaviors, such as inclination for one hop retweeting (*Finding 3*). In graph G, we consider each Weibo account i in our dataset as a node v_i. There is a directed edge e_{ij} from node v_i to node v_j, if the account i follows j. Furthermore, each following edge e_{ij} has a weight W_{ij}, determined by the retweeting similarity between each pair of accounts. The retweeting similarity quantifies the likelihood of the retweeting behavior. We use the Euclidean Distance to compute the similarity between a pair of nodes v_i and v_j based on the retweeting behavior. Retweeting vector R_i of node v_i is defined as follows:

$$R_i = \begin{bmatrix} x_1^i \\ x_2^i \\ \vdots \\ x_n^i \end{bmatrix} \tag{1}$$

where x_k^i is the times v_i retweets v_k's posts. $x_k^i = 0$ if v_i does not follow v_k or v_i follows v_k, but does not retweet v_k's posts. we use the following formula to calculate retweeting similarity (RS):

$$RS_{ij} = \frac{1}{1 + \sqrt{\sum_{k=1}^{n} (x_k^i - x_k^j)^2}} \tag{2}$$

Therefore, we can derive each following link's weight as follows:

$$W_{ij} = \frac{RS_{ij}}{\sum_{k \in followers(j)} RS_{kj}} \tag{3}$$

Algorithm 2 shows the SWIA, which has three phases: WR score initialization, WR score Propagation and WR score Aggregation.

WR score Initialization: We first assign an initial score d_i to each node v_i. Denote $S = \{S_i | S_i \text{ is a spam worker}\}$, then we assign $v_i \in S_i$ a non-zero score, $d_i = \frac{1}{|S|}$. For other accounts, the score is initialized to zero.

WR score Propagation: Our propagating procedure is similar to random-walk model, but with the following two modifications. First, the WR score is derived based on the score of a node's followees. As shown in Fig.9, user U_3 has one hop from user U_1, so its WR score derived from U_1 is $\frac{WR_{U_1}}{d_{U_1}}$, where d_{U_1} is the outdgree of U_1. For user U_4, it simultaneously follows U_1 and U_2. So its WR score

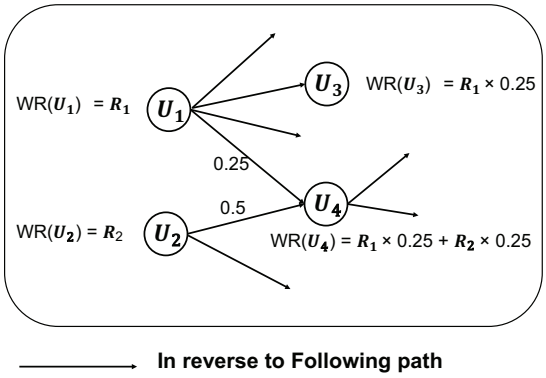


Fig. 9 WR score Propagation Procedure. (The direction of propagation is in reverse to the following relationship.)

is $\frac{WR_{U_1}}{d_{U_1}} + \frac{WR_{U_2}}{d_{U_2}}$. Therefore, a user who follows a large number of spammers get a high WR score. Second, the weight (W_{ij}) measures the similarity between two nodes i and j in terms of the retweeting behavior. W_{ij} is designed in such a way that these spam worker nodes will have a higher score than others after rounds of iterations.

WR score Aggregation: The effects of propagation mainly depend on the following relationships among users. Although we find that spam workers tend to be socially connected, there are still some exceptional spam workers like the nodes sparsely connected with the spammer community in the border area of Fig. 5. These exceptional spam workers just follow a few workers or even do not follow any other workers, showing a similar following behavior to normal users. That is, it is not possible for loosely connected spam workers to get a high WR score by only the propagation procedure. So the aggregation procedure is designed to address this issue. In the aggregation procedure, we measure the retweeting similarity between a non-seed node and every seed node by calculating the cosine similarity. Then we assign the Average Retweeting Similarity (ARS) score for each non-seed node:

$$ARS_i = \frac{1}{|S|} \sum_{j=1}^{|S|} \frac{R_i \cdot R_j}{||R_i||_2 ||R_j||_2} \tag{4}$$

where $|S|$ is the size of seed set and R_i is node v_i's retweeting vector. If we denote a user's WR score in propagation procedure as PWR, we can get a user's final WR score(WR) by setting two coefficient β and γ:

$$WR_i = \beta PWR_i + \gamma ARS_i \tag{5}$$

In Algorithm 2, at each iteration, we set an initial score bias. We set $\alpha = 0.85$, which is widely used in random-walk models. In this way, we can see that an account's PWR score can be proportionally distributed to its followers according to the closeness of social relationships and likelihood of their retweeting behavior. By summarizing the PWR score and ARS scores, the WR score can be obtained.

Algorithm 2 Spam Workers Inferring

Input: social graph, G; set of known spam workers, S; retweeting vecters, R; retweeting similarity weight matrix, W; initial score bias, α; coefficient, β and γ
Output: Worker Relevance (WR) score: WR
/*initialization score vector (d) for all nodes in G */
for each node i in set G
  if i in S
    $d_i = \frac{1}{|S|}$
  else
    $d_i = 0$
end for
/* compute Worker Relevance (WR) score */
$PWR \leftarrow d$
while PWR not converged **do**
  for each node i in G
    $t = \sum_{j \in followees(i)} PWR_j \times W_{ij}$
    $PWR_i = \alpha \times t + (1 - \alpha) \times d_i$
  end for
end while
for each node i in G

  $ARS_i = \frac{1}{|S|} \sum_{j=1}^{|S|} \frac{R_i \cdot R_j}{||R_i||_2 ||R_j||_2}$

  $WR_i = \beta PWR_i + \gamma ARS_i$
end for
return WR

6 Evaluation of Spammer Detection Algorithms

In this section, we evaluate our algorithms and compare them with other works. For SWIA, we compare it with the following algorithms:

Criminal account Inference Algorithm(CIA) [38]: A random-walk based algorithm. Impose semantic similarity of tweets to determine the weight of the folloing edges.

CollusionRank [12]: An algorithm adopts the ideas of spam-defense strategies proposed for the web gragh. The key idea is to penalize the users who follow a large number of spammers.

DetectVC [21]: A link-based algorithm to solve voluntary following problem. It incorporates both structure information of following behavior and prior knowledge collected from follower markets.

For SIIA, we compare it with the method used in [38] to infer the Criminal Hub and Leaf(CHL) as well as DetectVC. A major metric is the percentage of correctly inferred workers and initiators.

6.1 Dataset Selection

Our algorithms rely on known spammers. So we randomly choose a small set of 50 identified spam workers as our seed set (i.e. seed size $n = 50$). Then, using our crawlers, we obtain seed nodes' followee uids. Since it is impossible to collect all users' timelines, we randomly select about 50 accounts from each seed's followee list. Some of the accounts have expired or have been disabled by Weibo so that

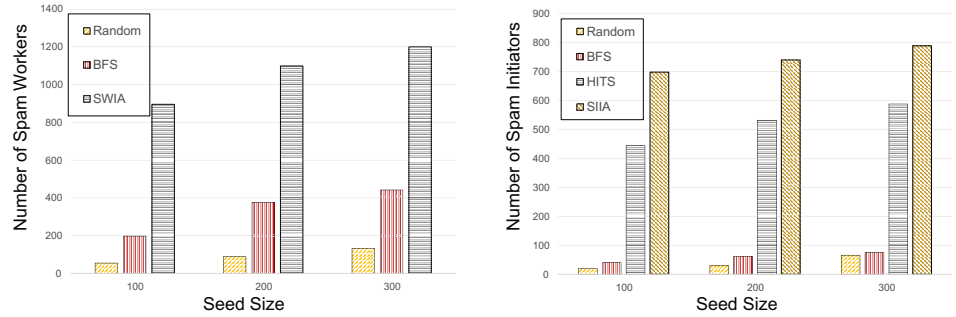
finally we get 63 spam workers and 237 spam initiators along with 1257 other users, which are treated as normal users. Therefore, we blend 50 identified spam workers (which will be used as seeds), their 1257 followees and 63 "unknown" spam workers and 273 "unknown" spam initiators together to obtain a set of 1607 accounts for evaluating our algorithms. For each user, we collect its timeline, which contains both original tweets and retweets. Then we extract all the edges of these nodes (accounts) and obtain a graph of 33,222 edges and 1,607 vertices. We also obtained three other datasets ($n = 100, 200, 300$) with different seed size through the similar method.

6.2 Inferring Spam Workers

We first compare SWIA with two intuitive methods: Random Selection and BFS (Breadth-First Search) using different seed size. In this experiment, starting from the same dataset generated from the same seed set, we use Random Selection, BFS and SWIA to infer spam workers respectively. Each algorithm outputs a sequence of accounts sorted by the WR score. Our purpose is to compare the performance of finding spam workers in top K nodes. As shown in Fig. 10(a), based on the number of seed spam workers, we can see our method outperforms the other two in different seed sets.

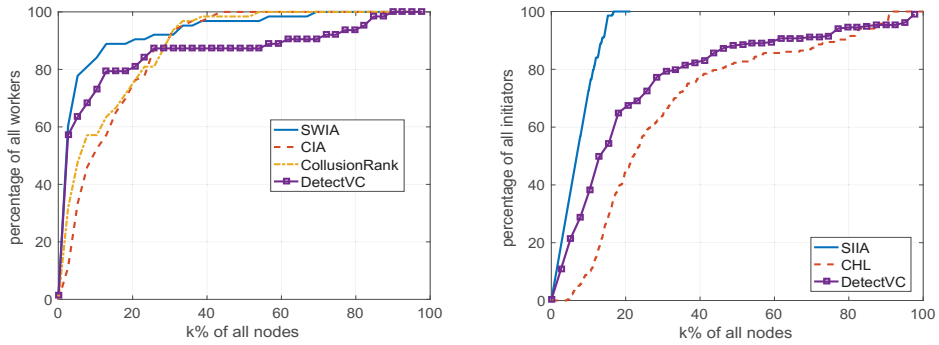
We also compare our algorithm with CIA, CollusionRank and DetectVC in terms of the percentage of known spam workers in top K of all ordered nodes by the WR score. The result is shown in Fig. 11 (a). We can see our algorithm is better than others since retweeting behavior is important in crowd-retweeting spamming, which is ignored in other three algorithms. Our algorithm SWIA finds 84% spam workers from top 10% users, while CIA finds 53%, CollusionRank finds 57% and 80% for DetectVC. However, an intersection between SWIA, CIA and CollusionRank occurs after finding 93% of all workers. The reason is that the spam workers whose rank is in the last 7% not only share similar retweeting behavior with normal users but also hold sparse following relationship with other workers. Therefore, our algorithm will have a lower WR score on these 7% workers than other algorithms which do not consider the retweeting behavior. Despite the intersection at the tail of the curve, our method has a good performance to identify potential spammers overall.

In order to take a deep look at the effects of PWR score and ARS score, we adjust coefficients β and γ and the result is shown in Fig.12 (a). Interestingly, with the increasing of γ, the front part of the curve goes higher while the tail part goes lower, which leads to an intersection of the curves under different β and γ. It means that before the intersection with the increasing of γ, the performance of detecting workers gets better, while after the intersection the bigger the β is, the better the performance is. So it can not achieve the optimal results to consider only one factor because the sum of β and γ equals 1. In our experiments, the coefficients β and γ are set to 0.4 and 0.6 respectively as a compromise.



(a) (b)

Fig. 10 (a)Performance of SWIA. (b)Performance of SIIA.

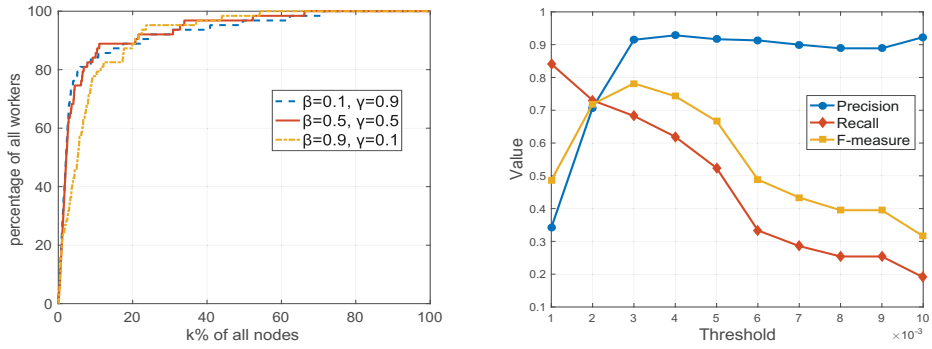


(a) (b)

Fig. 11 (a) SWIA vs CIA & CollusionRank & DetectVC in spam workers. (b) SIIA vs CHL & DetectVC in spam initiators.

6.3 Inferring Spam Initiators

We compare SIIA with two intuitive methods (BFS and Random) as well as the classic HITS. Similarly, we run these four algorithms on the same dataset derived from the same seed set. As shown in Fig. 10 (b), SIIA can always identify more spam initiators than other two methods. We also compare SIIA with DetectVC and CHL. In CHL a social graph is constructed considering only following links and uses HITS to rank the nodes by the hub score in a descending order. Fig. 11 (b) presents the percentage of identified spam initiators in the top K ranked nodes. We use the authority score to obtain the rank list since initiators are more likely to be followed by other users, especially by spam workers. When $K = 10$, SIIA using the authority score can find 70% initiators while CHL using the authority score finds only 15% and 40% for DetectVC. It can be observed that SIIA performs better than CHL and DetectVC since SIIA considers the retweeting behavior.



(a) (b)

Fig. 12 (a) SWIA under different β and γ. (b) SWIA under different thresholds.

Table 2 Comparisons of precision, recall and F-measure between different algorithms

Algorithms(worker)	Precision	Recall	F-measure
SWIA	0.915	0.683	0.782
CIA	0.284	0.397	0.331
CollusionRank	0.439	0.460	0.450
DetectVC	0.881	0.587	0.705
Algorithms(initiator)			
SIIA	0.875	0.886	0.881
CHL	0.327	0.688	0.443
DetectVC	0.543	0.667	0.598

6.4 Result Discussions

We adopt evaluation measures of precision, recall and F-measure to show the performance of these algorithms. For SWIA, we take WR score as threshold to divide positive and negative cases. Firstly the experimental results of SWIA with different thresholds are shown in Fig. 12 (b). We can see that our algorithm SWIA gets its highest F-measure value when the threshold is set to 3.0×10^{-3}. The thresholds of other algorithms are figured out by the same experimental method. The comparisons of precision, recall and F-measure are shown in Table 2. SWIA can achieve better performance in spam worker detection. However, CIA gets the lowest F-measure because crowdsourcing spammers usually consist of many normal-like users, which means their posted tweets are manually edited instead of generated by scripts. So the semantic similarity used in CIA cannot work on those normal-like spam workers. CollusionRank and DetectVC's performance is barely satisfactory since these two link-based methods have not considered the retweeting behavior of users, which is a significant feature in crowd-retweeting based spamming. The result in Table 2 also indicates the effectiveness of retweeting topology since SIIA performs better than CHL and DetectVC. Both CHL and DetectVC impose following relationship in the detection of initiators while it is not so useful when the majority of initiators' following pattern is similar to normal users.

7 Conclusion

In this paper, we investigate spamming in a crowd-retweeting system in social networks. Based on our reliable data crawled from ZBJ, SDH and Weibo, we find spammers are closely connected and form a community and spam workers are more likely to follow each other. Spam workers incline to retweet the original messages posted by initiators due to the crowd-retweeting spamming task's structure. Based on the analysis and observations, we design two algorithms to infer spam workers and initiators using a set of seed spam accounts. In practice, we can use the two algorithms iteratively to find more suspicious spamming accounts.

Acknowledgements This work is supported by National Natural Science Foundation of China under Grants, No. 61370208, No. 61472081, No. 61772133, No. 61402104, No. 61320106007, No. 61370207, US NSF under awards CNS-1527303 and OAC-1642124, Collaborative Innovation Center of Wireless Communications Technology, Collaborative Innovation Center of Social Safety Science and Technology, Jiangsu Provincial Key Laboratory of Network and Information Security (BM2003201),and Key Laboratory of Computer Network and Information Integration of Ministry of Education of China under Grants No. 93K-9.

References

1. A. H. Wang, A.H.: Don't follow me - spam detection in twitter. In: Proc. of IEEE SECRYPT, pp. 142–151 (2010)
2. Becchetti, L., Castillo, C., Donato, D., Baeza-YATES, R., Leonardi, S.: Link analysis for web spam detection. ACM Trans. Web **2**(1), 2:1–2:42 (2008)
3. Benevenuto, F., Magno, G., Rodrigues, T., Almeida, V.: Detecting spammers on twitter. In: In Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS) (2010)
4. Chen, C., Wen, S., Zhang, J., Xiang, Y., Oliver, J., Alelaiwi, A., Hassan, M.M.: Investigating the deceptive information in twitter spam. Future Generation Computer Systems **72**, 319–326 (2017)
5. Chen, C., Wu, K., Srinivasan, V., Zhang, X.: Battling the internet water army: Detection of hidden paid posters. In: Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM '13, pp. 116–120 (2013)
6. Crawler4j: https://github.com/yasserg/crawler4j (2017)
7. C.Smith: 61 amazing weibo statistics and facts. Digital Statistics and Gadgets 2017-03-18 (2017). URL http://expandedramblings.com/index.php/weibo-user-statistics/
8. Danezis, G., Mittal, P.: Sybilinfer: Detecting sybil nodes using social networks. Proceedings of the Network and Distributed System Security Symposium(NDSS), San Diego, California, USA, 8th February - 11th February (2009)
9. Fayazi, A., Lee, K., Caverlee, J., Squicciarini, A.: Uncovering crowdsourced manipulation of online reviews. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 233–242. ACM (2015)
10. Fielding, N., Cobain, L.: Revealed: Us spy operation that manipulates social media. http://www.guardian.co.uk/technology/2011/mar/17/us-spy-operation-social-networks (2011)
11. Freelancer.com: https://www.freelancer.com/about (2017)
12. Ghosh, S., Viswanath, B., Kooti, F., Sharma, N.K., Korlam, G., Benevenuto, F., Ganguly, N., Gummadi, K.P.: Understanding and combating link farming in the twitter social network. In: Proceedings of the 21st International Conference on World Wide Web, WWW '12, pp. 61–70 (2012)
13. Gyöngyi, Z., Garcia-Molina, H., Pedersen, J.: Combating web spam with trustrank. In: Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30, VLDB '04, pp. 576–587 (2004)
14. I.Lunden: Twitter vulnerability alllows cyber criminals to spread spam. http://www.one.com (2012)

15. Kittur, A., Chi, E.H., Suh, B.: Crowdsourcing user studies with mechanical turk. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '08, pp. 453–456 (2008)
16. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. J. ACM **46**(5), 604–632 (1999)
17. Kwak, H., Lee, C., Park, H., Moon, S.: What is twitter, a social network or a news media? In: Proceedings of the 19th International Conference on World Wide Web, WWW '10, pp. 591–600 (2010)
18. Lee, K., Caverlee, J., Webb, S.: Uncovering social spammers: Social honeypots + machine learning. In: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '10, pp. 435–442 (2010)
19. Li, H., Chen, Z., Mukherjee, A., Liu, B., Shao, J.: Analyzing and detecting opinion spam on a large-scale dataset via temporal and spatial patterns. In: ICWSM, pp. 634–637 (2015)
20. Liu, B., Luo, J., Cao, J., Ni, X., Liu, B., Fu, X.: On crowd-retweeting spamming campaign in social networks. In: IEEE International Conference on Communications (ICC), 2016., pp. 1–6. IEEE (2016)
21. Liu, Y., Liu, Y., Zhang, M., Ma, S.: Pay me and i'll follow you: Detection of crowdturfing following activities in microblog environment. In: International Joint Conference on Artificial Intelligence (IJCAI) (2016)
22. Matthias, H., Tobias, H., Tran-Gia, P.: Anatomy of a crowdsourcing platform - using the example of microworkers.com. In: Proceedings of the 2011 Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IMIS '11, pp. 322–329 (2011)
23. Microworkers.com: https://ttv.microworkers.com/index/template (2017)
24. Motoyama, M., McCoy, D., Levchenko, K., Savage, S., Voelker, G.M.: Dirty jobs: The role of freelance labor in web service abuse. In: Proceedings of the 20th USENIX Conference on Security, SEC'11, pp. 14–14 (2011)
25. Sandaha: http://www.sandaha.com/ (2017)
26. Scrapy: https://github.com/scrapy/scrapy (2017)
27. SDK, S.W.O.: http://open.weibo.com/wiki/SDK (2017)
28. Sparks, D.: How many users does twitter have? The Motley Fool (2017)
29. Thomas, K., Grier, C., Song, D., Paxson, V.: Suspended accounts in retrospect: An analysis of twitter spam. In: Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference, IMC '11, pp. 243–258 (2011)
30. Tran, N., Min, B., Li, J., Subramanian, L.: Sybil-resilient online content voting. In: Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation, NSDI'09, pp. 15–28 (2009)
31. Turk, A.M.: https://requester.mturk.com/ (2017)
32. Twitter: http://www.twitter.com/ (2017)
33. Viswanath, B., Post, A., Gummadi, K.P., Mislove, A.: An analysis of social network-based sybil defenses. SIGCOMM Comput. Commun. Rev. **41**(4) (2010)
34. Wang, G., Wilson, C., Zhao, X., Zhu, Y., Mohanlal, M., Zheng, H., Zhao, B.Y.: Serf and turf: Crowdturfing for fun and profit. In: Proceedings of the 21st International Conference on World Wide Web, WWW '12, pp. 679–688 (2012)
35. Weibo, S.: http://weibo.com/ (2017)
36. Wikipedia: Internet water army. http://en.wikipedia.org/wiki/Internet_Water_Army (2015)
37. Xue, J., Yang, Z., Yang, X., Wang, X., Chen, L., Dai, Y.: Votetrust: Leveraging friend invitation graph to defend against social network sybils. In: Proceeding of The 32nd IEEE International Conference on Computer Communications, INFOCOM '2013 (2013)
38. Yang, C., Harkreader, R., Zhang, J., Shin, S., Gu, G.: Analyzing spammers' social networks for fun and profit: A case study of cyber criminal ecosystem on twitter. In: Proceedings of the 21st International Conference on World Wide Web, WWW '12, pp. 71–80 (2012)
39. Yang, C., Harkreader, R.C., Gu, G.: Die free or live hard? empirical evaluation and new design for fighting evolving twitter spammers. In: Proceedings of the 14th International Conference on Recent Advances in Intrusion Detection, RAID'11, pp. 318–337 (2011)
40. Yang, X., Yang, Q., Wilson, C.: Penny for your thoughts: Searching for the 50 cent party on sina weibo. In: ICWSM, pp. 694–697 (2015)
41. Yang, Z., Wilson, C., Wang, X., Gao, T., Zhao, B.Y., Dai, Y.: Uncovering social network sybils in the wild. In: Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference, IMC '11, pp. 259–268 (2011)

42. Yu, H., Gibbons, P.B., Kaminsky, M., Xiao, F.: Sybillimit: A near-optimal social network defense against sybil attacks. IEEE/ACM Trans. Netw. **18**(3), 885–898 (2010)
43. Yu, H., Kaminsky, M., Gibbons, P.B., Flaxman, A.: Sybilguard: Defending against sybil attacks via social networks. In: Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM '06, pp. 267–278 (2006)
44. Zhubajie: http://www.zhubajie.com/ (2017)