On Crowd-Retweeting Spamming Campaign in Social Networks

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

Benyuan Liu, Xinwen Fu
Department of Computer Science
University of Massachusetts Lowell, Mansasuetts, USA
Email:{bliu, xinwenfu}@cs.uml.edu

***Abstract*—Crowdsourcing is often used to solicit contributions from an online community for ideas, evaluation and opinions. However, spamming can pollute such a system and manipulate the results of crowdsourcing. For detection of those spammers, the training data used in previous studies is often derived by experts labeling collected data and manually identifying spammers. The reliability of such training data is questionable. In this paper, we utilize two web based service providers Zhubajie (ZBJ) and Sandaha (SDH) and obtain reliable data about the spammers. We use such data to investigate the crowd-retweeting spam in Sina Weibo. We analyze profile features, social relationship and retweeting behavior of such spammers. We find that although these spammers are likely to connect more closely than legitimate users, the underlying social tie is different from the social relationship in other spam campaigns because of the unique retweeting features with the information cascade effect. Based on these findings, we propose retweeting-aware link based ranking algorithms to detect suspect spam accounts using seeds of identified spammers. Our evaluation shows that our algorithm is more effective than other link-based methods.**

***Index Terms*—Social Network; Crowd-Retweeting; Spamming;**

I. Introduction

Spam campaigns have been polluting various crowdsourcing systems. It was reported that the US military developed specific software to speed up the distribution of pro-American propaganda in social media [1]. In China, "Internet Water Army" is a group of paid writers posting on social media to advertise or manipulating the public opinion [2], [3]. 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 retweets spread widely and quickly.

In this paper, we investigate the crowd-retweeting based spam in the Sina Weibo (Weibo) social network. We crawled a large dataset of Weibo accounts based on a large set of spammer accounts, extracted from two public web based service providers - Zhubajie (ZBJ) [4] and Sandaha (SDH) [5], two representative crowdsourcing systems [6]. We carefully study and analyze profile features, social relationship and retweeting behavior of such spammers. We statistically analyze the characteristics of such a water army to understand their behavior. We propose two inferring algorithms to find more suspect spam initiators and workers respectively. The experiment results show that our methods perform well to identify potential spammers.

The main contributions of this paper are summarized as follows: 1. To the best of our knowledge, we are the first to analyze the social structure of crowdsourcing based spammer community and examine the two roles of spammers: initiator and worker. We collected reliable training data about spammers from Zhubajie (ZBJ) and Sandaha (SDH). 2. We analyze spammer characteristics and discover the unique features of spam initiators and workers. Our findings show a spam initiator acts like other non-spammmers except for its way of retweeting. In contrast, spam workers are more closely connected among themselves with strong social relationship and their retweeting behavior is different from normal users. 3. We design two inferring algorithms to find more suspect spammers based on both link structure and retweeting features.

The rest of this paper is organized as follows. We present related work in Section II. Section III introduces how to collect real-world data. Section IV provides the detailed analysis of spammers' characteristics. Section V presents our algorithms to detect such spammers. Evaluation results are presented in Section VI. Section VII concludes this paper.

II. Related Work

Crowdsourcing systems [7] [8] have been widely studied [9], [10]. Abuse of crowdsourcing systems has attracted attention recently. Motoyama *et al.* [11] characterize the abuse-related labor on Freelance.com. Chen *et al.* [3] analyzed the hidden paid posters and proposed a SVM based detection scheme using posters' behavior features and postings' semantic similarity. Wang *et al.* [6] confirmed the existence of malicious usage of crowdsourcing systems by analyzing two crowdturfing websites ZBJ and SDH. They compared five types of campaigns. Particularly they checked suspect accounts' profiles and crowdturfing's effect on information dissemination in Weibo. However, their analysis doe not consider the underlying social structure of the social network and no detection method is proposed.

Recently, researchers began to pay attention to a spammer community's social structure. Yang *et al.* [12] found malicious users are socially connected and proposed an inference algorithm based on identified malicious accounts. They used malicious URLs to define spammers, which are not crowdsourcing spammers studied in this paper. Ghosh *et al.* [13] investigated the link farm in Twitter and presented a

978-1-4799-6664-6/16/$31.00 ©2016 IEEE

scheme using links to reduce the rank of malicious users. However, they consider only *following* links to reduce the page rank as a search engine does. It is already found [14] that ranking by *following* relationship and retweeting is different. Therefore, the above methods is not sufficient for detecting crowd-retweeting spammers.

In this paper, our data collection strategy is quite different from those used in related work. We utilize data of real paid posters as our seeds to crawl more data. We find that although spammers are connected closely, their relationship is often unidirectional. Spammers do not just gain influence by *following* and they also spread spams by retweeting. This makes the strategy of using only the *following* link to detect spammers not sufficient.

III. Data collection in Weibo

A crowd-retweeting spam campaign has three key actors: **1. Customer**: An individual or a company who initiates a spam campaign. The customer is the spam initiator and pays for the cost. For example, a customer can initiate a campaign by posting on Weibo. **2. Worker**: Paid posters that perform tasks assigned by customers. In Weibo, they are spam workers who retweet or comment on tweets of a customer to attract other normal Weibo users in exchange for a fee. **3. Agent**: Intermediaries who take charge of task finding, managing and distributing funds to workers to accomplish the goals. In our data, ZBJ and SDH are web based services acting as agents. Agents can be in different forms such as websites and forums. For example, a customer can hire spam workers from ZBJ or SDH, asking these workers to distribute her post on Weibo.

We collected two datasets for the study of spammers in a spcrowd-retweeting spam campaign:

***Data set 1*. A large number of identities of 14,443 real Weibo Spammers** from crowdsourcing systems Zhubajie (ZBJ) and Sandaha (SDH), where they get paid by retweeting messages in the Weibo social network. ZBJ and SDH were used in a previous work [6], which did not explore the social relationship link structure and retweeting behavior. We developed a web crawler based on crawler4j [15] and crawled ZBJ and SDH webpages that contain Weibo's spam tasks. These spammer data from ZBJ and SDH can be used for finding spammer characteristics and also work as seeds for finding other spammers. We also crawled Weibo and obtained the profile, follower's uid list, *following*'s uid list, tweets and retweets of these spammers found in ZBJ and SDH. The data includes 32.6GB webpages from both ZBJ and SDH and was obtained in two weeks in October 2012.

***Data set 2*. A large dataset of 193,591 Weibo user accounts**, which are treated as normal Weibo accounts and used for comparison with spammers. We crawl Weibo with the Sina Weibo Open SDK [16] and obtain Weibo user information including uid, profile, a follower's uid list, *following*'s uid list, tweets and retweets. This dataset has 193,591 users and 10,785,921 tweets. From this large dataset, we randomly picked up five small samples for comparison with spammers from the large data set. Each sample has around 1500 Weibo accounts. The data was obtained in 2012.

IV. Analysis of Crowd-retweeting Spamming

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

A. Profile Characteristics

Our goal is to find features of spammers in order to identify them. We first try to obtain such features from profiles of Weibo accounts. The profiles can provide features such as the number of followers, *following* and tweets. These statistics often reflect reputation of Weibo accounts in the social network. They are often used as features by spam detection algorithm [17], [18]. To verify whether these features are still useful, we compare the profiles of spammers with profiles of normal users in the five normal user samples. Compared with normal users, spammers have more *followings* and followers. For example, 80% of normal users follow no more than 500 users, while only 40% of spammers follow fewer than 500 users. 90% of normal users, compared with 35% of spammers, has no more than 1,000 followers. Most spammers post fewer tweets than normal users. Therefore, the number of *followings* and followers can differentiate spammers from normal users to some extent, particularly spam workers.

B. Social Relationship Characteristics

We perform various statistical analysis to identify social relationship features of spammers in Weibo. Below are our findings.

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 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 relationship 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. We firstly calculate the reciprocated vertex pair (RVP) ratio for the spammer graph and the five normal user samples. Figure 1 (a) is the empirical cumulative distribution function (ECDF) of RVP ratio of normal user samples. Figure 1 (b) shows that the RVP ratio distribution of spammers (including both workers and initiators) is similar to the distribution of normal users.

From the analysis above, it can be observed that spammers connect closely to form a community in the social network. 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 subgraph. Then we compute and analyze the RVP ratio of spammers. The results are shown in Figure 1 (c). For the spam workers, we find 70% of the RVP radios 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. Spam initiators behave more like normal users.

C. 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. Figure 2 (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 counter intuitive since intuitively spam workers are more likely to retweet in order to make money.

We perform the following analysis to find the reason why more spam workers tweet less and have observed unique features of spammers. There are two types of retweeting by a user of interest: retweeting posts created by accounts in the *following* list of the user of interest and retweeting posts created by other accounts rather than accounts in the *following* list of the user of interest. Figure 2 (b) is the ECDF of the number of retweeted posts that are created by accounts in the *following* list of normal users and spam workers versus number of *followings* whose posts are retweeted. We denote *accounts in the following list* as *followings* for brevity. 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 *followings* than normal users. We go further to analyze the one-hop retweeting behavior. Figure 2 (c) gives the ECDF of the number of followings whose posts are retweeted only once by spam workers. It can be observed that only 10% of spammers and normal users retweet posts of more than 50 *followings*. However, it is less likely that retweeted posts from spam workers are retweeted again by the followers of the spam workers.

V. Algorithms of Detecting Spammers

The basic idea of finding spammers is to search for suspect 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.

A. Inferring Spam Initiator

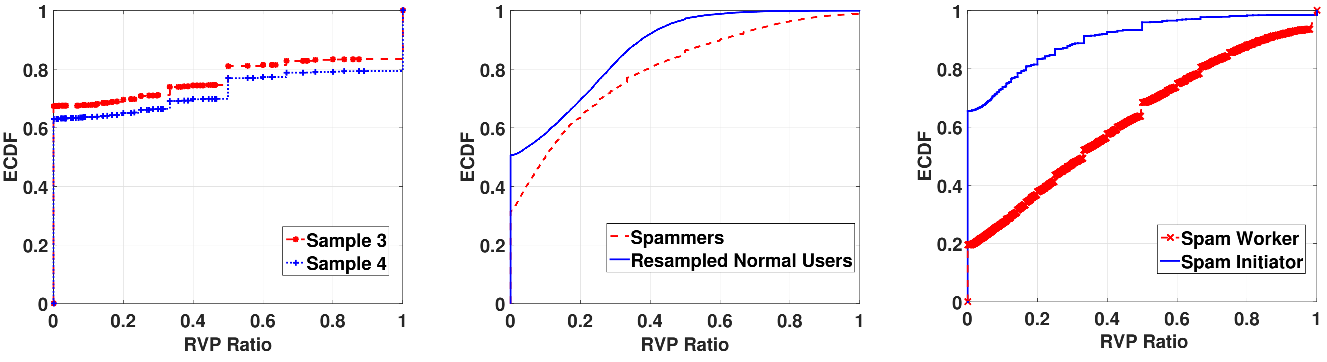
Our Spam Initiator Inference Algorithm (SIIA) uses a similar strategy to the HITS [19] algorithm in order to find the spam initiators. In Section IV, we find that a worker is more likely to become the follower of an initiator than other social network users and the worker inclines to forward or comment on an initiator's tweets than other *followings* of the worker. Therefore, an initiator can be viewed as an authority with many incoming links while spam workers are hubs with many outgoing links. Our SIIA algorithm is different from the HITS algorithm since we consider only the one hop retweeting (*following*) relationship given our observations in Section IV, excluding multi-hop retweeting edges.

Algorithm 1 is the sketch of the 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. We first construct a new graph G' according to the seed node's one hop retweeting list and *following* links and then use the HITS algorithm to 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 suspected initiator nodes and select top K users with a high authority score as the spam initiators.

B. Inferring Spam Worker

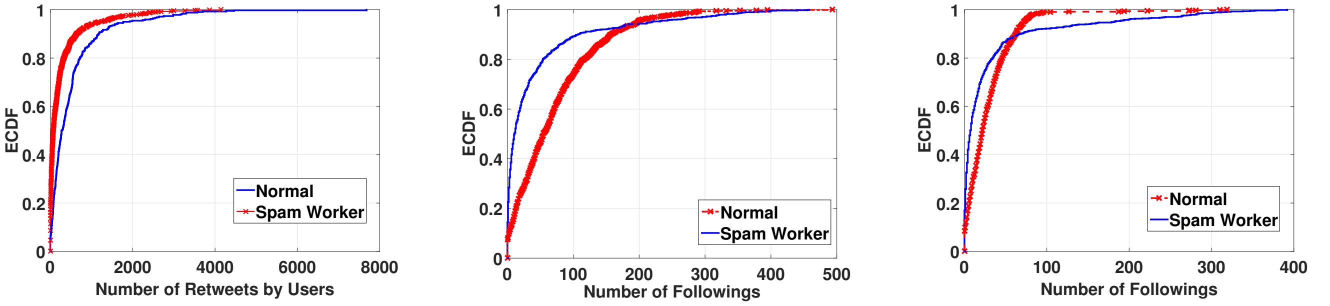
Spam workers are more closely connected than spam initiators. We design a Spam Worker Inference Algorithm (SWIA), which propagates the spam relevance scores from a seed set of real spammer accounts to their *followings*. If an account has a high spamming score, it will have a high rank in our spammer list. Our algorithm is based on two findings in Section IV: (1) Spam Workers tend to be socially connected; (2) Spam Workers usually share similar tweeting behaviors, such as inclination for one hop retweeting. We build a social graph $G = (V, E)$ to model the behavior of spam workers. In this graph, we consider each Weibo account i in our dataset as a node v_i. There is a directed edge e_{ij} from the node v_i to the node v_j, if the account i follows j.

Our basic idea is to infer spam workers from a set of identified spam workers. We introduce Spam Worker Relevance Score to measure a user's relevance with the spam worker.



(a) (b) (c)

Fig. 1: RVP Ratio for (a) Sample 3 vs Sample 4. (b) Resample vs Spammers. (c) Spam Initiator vs Spam Worker.



(a) (b) (c)

Fig. 2: (a) ECDF of the number of retweets by users (b) ECDF of the number of retweeted posts that are created by accounts in the *following* list (c) ECDF of the number of retweeted posts that are created by accounts in the *following* list versus number of *followings* whose posts are retweeted only once.

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 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. Vector r_i is defined as follows,

$$r_i = \begin{bmatrix} x_1^i & x_2^i & \cdots & x_n^i, \end{bmatrix} \tag{5}$$

where x_k^i is how many 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 :

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

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

$$W_{ij} = \frac{RS_{ij}}{\sum_{k \in followers(j)} RS_{kj}} \tag{2}$$

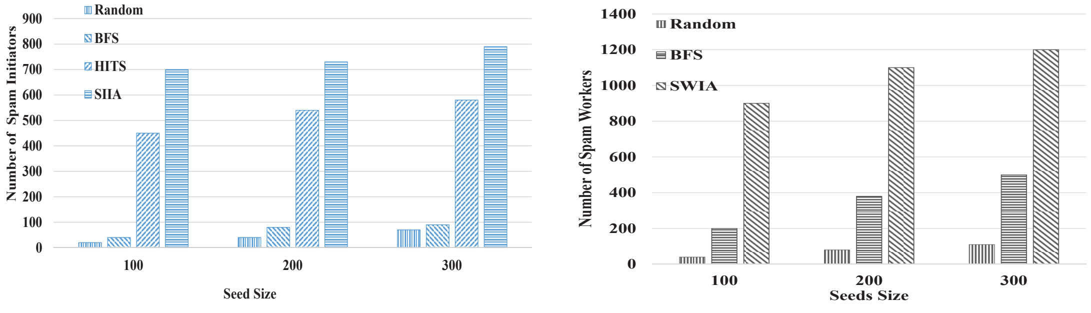
Algorithm 2 shows the SWIA, which has two phases: Spam Worker Relevance Score initialization and Spam Relevance (SR) Score Propagation. *SR Score Initialization*: Denote $S = \{S_i | S_i$ is an indentified spam worker$\}$. We assign $v_i \in S$ a non-zero score, $d_i = \frac{1}{|S|}$. For other nodes (users), the score is initialized to zero. *SR Score Propagation*: Our propagating algorithm is different from the one used by the PageRank algorithm, as follows. First, the SR score is derived based on the score of a node's *followings*. Therefore, a user who is followed by a large number of spammers get a high SR score. Second, the weight $(W(i,j))$ measures the similarity between two nodes i and j in terms of the retweeting behavior. $W(i,j)$ is designed in such a way that these spam worker nodes will have a higher score than others after rounds of iterations.

VI. Evaluation of Spammer Detection Algorithms

In this section, we evaluate our algorithms and compare their performance with related work. For SWIA, we compare it with Collusionrank [13]. For SIIA, we compare it with the method used in [12], which infers the criminal hub and leaf (CHL).

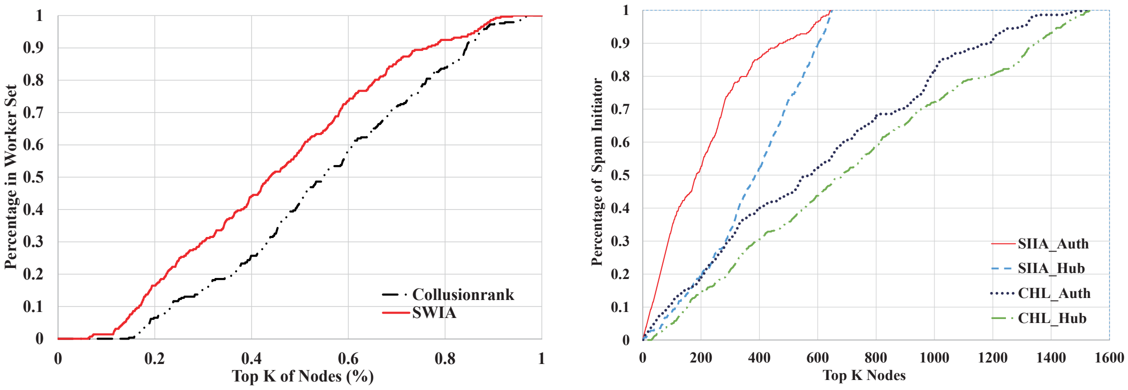
A. Dataset Selection

For the purpose of evaluation, we randomly choose a small set of 50 identified spam workers as our seed set.



(a) (b)

Fig. 3: (a) Performance of SIIA. (b) Performance of SWIA.



(a) (b)

Fig. 4: (a) SWIA vs Collusionrank in worker. (b) SIIA vs CHL in initiator.

We obtain 16 *followings* of each spammer worker and these *followings* were known as spammers. Some accounts of the *followings* expired or were disabled by Weibo so that we derived 760 *followings*. We also obtained 292 spam workers and 432 initiators who were also *followings* of the 50 identified spam workers. Therefore, we blend the 50 identified spam workers, their 760 *followings* and the 292 spam workers and 432 initiators together to obtain a set of 1,534 accounts for evaluating our algorithms. We also extract all the edges of these nodes (accounts) and obtain a graph of 11,350 edges and 1,534 vertices. Similarly, we obtained social graphs when the size of seed nodes were 100, 200, and 300.

B. *Inferring Spam Workers*

We first compare SWIA with two intuitive methods: Random Selection and BFS (Breadth-First Search) using different seed size. As shown in Figure 3 (b), SWIA can always identify more spam workers than other two methods. We also compare our algorithm with Collusionrank in term of what percentage of spammer workers and initiators are found if we choose top K nodes that are ranked by the value of SR score. The result is shown in Figure 4 (a). We can see our algorithm is better than Collusionrank that considers only the *following* links. When $K = 97$, SWIA finds the first spam worker that is not a seed spam worker, but a sapmmer in the set of known 292 spam workers and 432 initiators, while the Collusionrank finds the first non-seed spammer worker when $K = 225$. Actually, when $K = 96$, SWIA finds all 50 seed workers.

C. *Inferring Spam Initiators*

We compare SIIA with two intuitive methods (BFS and Random) as well as classic HITS. As shown in Figure 3(a), SIIA can always identify more spam initiators than other two methods with different seed set size. We also compare SIIA with CHL, in which a criminal leaf is a spam initiator and a criminal hub is a worker. The CHL constructs a social graph considering only *following* links and uses HITS to rank the nodes by the hub score in a descending order. Figure 4 (b) presents the percentage of identified spam initiators versus the first K ranked nodes. Four algorithms are compared: SIIA_Auth (SIIA using the authority score), SIIA_Hub (SIIA using the hub score), CHL_Auth (CHL using the authority score), CHL_Hub (CHL using the Hub score). It can be observed that using the authority score is much better than using the hub score, since initiators are more likely to followed by other users, especially by spam workers. It can be observed that SIIA performs better than CHl since SIIA considers the retweeting behavior. When $K = 75$, SIIA using the authority score can find 70 initiators while CHL using the authority score finds only 35.

Algorithm 1 Spam Initiator Inferring

```
Input: social graph, $G = (V, E)$; set of known spammers, $S$; user's one hop retweet list, $L$
Output:Spam Initiator Relevance Score: s
$G' = \emptyset$
for each user $u$ in set $S$
    add $u$ as a vertex to $G'$
    for each node $v$ in $u_i$'s retweeted 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 $b$ for $G'$ */
/*initialize score vector $d_a$ and $d_b$ for all nodes $n$ in $G'$*/
$d_a(n) = 1$, $d_b(n) = 1$
/*compute spam initiator relevance scores, nf denotes the followings of node $n$*/
while $d_a$ or $d_b$ not converged do
    for all nodes $n$ in $G'$ do
        $d_b(n) = \sum_{nf \in followings(n)} d_a(nf)$
        $d_a(n) = \sum_{nfr \in followers(n)} d_b(nfr)$
    end for
    Normalise the hub and authority scores
end while
return $s = d_a$
```

Algorithm 2 Spam Workers Inferring

```
Input: social graph, $G$; set of known spam workers, $S$; decay factor, $\alpha$; retweeting similarity weight matrix, $W$
Output: Spam Worker Relevance (SR) Score: $sr$
/*initialization score vector ($d$) for all nodes in $G$ */
for each node $i$ in set $S$
    if $i$ in $S$
        $d(i) = \frac{1}{|S|}$
    else
        $d(i) = 0$
end for
/* compute Spam Worker Relevance score */
$sr \leftarrow d$
while $sr$ not converged do
    for each node $i$ in $G$
        $t = \sum_{j \in followings(i)} sr(j) \times W(i, j)$
        $sr(i) = \alpha \times t + (1 - \alpha) \times d(i)$
    end for
end while
return $sr$
```

VII. Conclusion

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

Acknowledgement

This work is supported by National Natural Science Foundation of China under Grants, No. 61370208, No. 61472081, No. 61320106007, No. 61272531, China National High Technology Research and Development Program (2013AA013503), NSF under award CNS-1527303, Collaborative Innovation Center of Wireless Communications 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] N. Fielding and L. Cobain, "Revealed: Us spy operation that manipulates social media." http://www.guardian.co.uk/technology/2011/mar/17/us-spy-operation-social-networks, 2011.

[2] "Internet water army." http://en.wikipedia.org/wiki/Internet_Water_Army, 2015.

[3] C. Chen, K. Wu, V. Srinivasan, and X. Zhang, "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.

[4] "Zhubajie." http://www.zhubajie.com, 2015.

[5] "Sandaha." http://www.sandaha.com, 2015.

[6] 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 *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pp. 679–688, 2012.

[7] "Freelancer.com." http://www.freelancer.com/info/about.php, 2015.

[8] H. Matthias, H. Tobias, and P. Tran-Gia, "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.

[9] "Amazon mechanical turk." http://Aws.amazon.com/cn/mturk, 2015.

[10] A. Kittur, E. H. Chi, and B. Suh, "Crowdsourcing user studies with mechanical turk," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pp. 453–456, 2008.

[11] M. Motoyama, D. McCoy, K. Levchenko, S. Savage, and G. M. Voelker, "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.

[12] C. Yang, R. Harkreader, J. Zhang, S. Shin, and G. Gu, "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.

[13] S. Ghosh, B. Viswanath, F. Kooti, N. K. Sharma, G. Korlam, F. Benevenuto, N. Ganguly, and K. P. Gummadi, "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.

[14] H. Kwak, C. Lee, H. Park, and S. Moon, "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.

[15] "Crawler4j." https://github.com/yasserg/crawler4j, 2015.

[16] "Sina weibo open sdk." http://open.weibo.com/wiki/SDK, 2015.

[17] A. H. Wang, "Don't follow me - spam detection in twitter.," in *Proc. of IEEE SECRYPT*, pp. 142–151, 2010.

[18] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida, "Detecting spammers on twitter," in *In Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS)*, 2010.

[19] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *J. ACM*, vol. 46, pp. 604–632, Sept. 1999.