Mining Potential Spammers from Mobile Call Logs

Zhipeng Liu, Dechang Pi, and Yunfang Chen

1 College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 210016, China
2 Department of Software Engineering, Nanjing University of Posts and Communications, Nanjing, Jiangsu 210003, China

Received 20 June 2014; Revised 20 August 2014; Accepted 9 September 2014

1. Introduction

Voice call has become a basic and reliable service provided by mobile network operators for many years. Although it has been partly replaced by many applications on smartphones, such as LINE and WeChat, it is still the most widely used mobile communication method. With the rapid growth of mobile telecommunication, more and more people in China are equipped with mobile phones [1, 2]. However, the number of spam calls has increased dramatically (http://www.shanghaidaily.com/metro/society/Harassing-phone-calls-from-city-plague-2m/shdaily.shtml). Many users have become the victims of phone scams and suffered heavy financial loss (http://www.hollywoodreporter.com/news/chinese-actress-tang-wei-falls-670430). The content of spam calls can be classified into three categories: harassment, fraud, and illegal advertisement. The harassment calls are composed of jokes and hoaxes. The consequence is often harmless. However, the fraud and illegal advertisement calls can be very harmful. These malicious calls induce people to pay money for unwanted subscription, transfer money, and expose serious personal information. Many mobile phone users trust the content of spam calls and the response rates are high compared with traditional Email spam. It is necessary to detect potential call spammers as soon as possible.

There are two popular methods for call spam detection: the server-based solution provided by cellular network operators and the client-based third-party applications provided by software companies. Since it is huge workload to analyze call contents at the server side, in reality, the task of call spam detection is mainly performed by applications on smartphones, such as Tencent mobile manager. However in China, there are 70 percent of mobile users in rural areas who still use feature phones, which cannot install antispam software and thus have completely no protection against call spam. There are also two popular methods for call spam detection: the crowdsourcing approach and the automatic call spam filtering. In reality, the crowdsourcing approach is widely adopted by client-based applications. People can label a specified phone number as spammer by software, and then this information is sent to a server and distributed to other users as needed. However, research results have shown that almost all third-party applications do not use machine learning.
learning methods, thus the effectiveness of antispam software have been restricted [3]. The automatic method takes call log into consideration, uses a social network analysis approach and analyzes the behavior patterns of mobile users. This method has the advantage of being deployed on the server of mobile network operators. Based on the analysis above, we summarize that it is necessary to perform call spam detection at the server side automatically with minimum user information.

In this study, we try to conduct a systematic investigation of the case of finding potential call spammers from mobile call logs. Identification of call spammers can provide better mobile experience, as it protects people against malicious information senders.

The approach we take to handle call spam detection is rather different from existing relevant research.

1) Log Information Only. Our work only uses call log information. There is no supervised information indicating which node sends call spam. However, the content-based methods use the content of the call as a judgment of ham or spam. Our method protects user privacy, while achieves good performance.

2) Time-Dependent. The role of a node in mobile network may shift dynamically. One could turn from a call spammer to a normal node and vice versa. However, there is no clear sign when the transition happens.

3) Server-Based Solution. Most of the existing methods utilize client-based applications. Our method can be deployed on the servers of cellular network operators and thus can provide protection for smartphones as well as feature phones.

In this paper, we formulate the problem of mining call spammers as a classification problem and propose a time-based communication graph to model the dynamic communication network. Our model partitions mobile communication stream into a series of segments, builds the time-based communication graph, then uses an algorithm with ranking and classification to detect the potential call spammers. Experimental results have shown that this unsupervised approach can achieve an accuracy of 84.5–91.8%. Our method can detect call spammer automatically, dynamically and effectively with minimum user information.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 motivates the call spammer detection problem. Section 4 describes the proposed CLRank algorithm. Section 5 presents experimental results and compares the effectiveness of CLRank with other baseline methods. Section 6 concludes the study.

2. Related Work

In this section, we discuss several areas of related work: (1) complete manual methods which are used by cellular network operator, (2) semiautomatic content-based call spam filtering methods, and (3) automatic social network call spam filtering methods.

1) The most widely used methods by cellular network operator are the restriction of maximum number of calls per day. However, the restriction method taken by cellular network operator has the following disadvantages: (1) it is hard to define the threshold of number of calls per day. Some public service numbers may make a lot of phone calls in one day legally; (2) phone scam criminals can use different phone numbers to perform spam calls, while the calls of each number per day is within the threshold.

2) The content-based call spam filtering techniques adopted by client-based applications are based on crowdsourcing and phone number blacklisting. For example, Yadav et al. propose a client-based Naïve Bayes filter for user to send feedback to a centralized server, which can further benefit other users [4]. The effectiveness of the system is largely depends on user feedback. However, many people do not send feedback in time, the effectiveness of the system is relatively low. In China, there are Tencent mobile manager and Qihoo 360 Mobile Security Solution which have similar functionalities.

3) Social network spam detection approaches are mainly based on the fundamental technological advancement of collecting, organizing and storing network data [5–11]. Sun and Jara have proposed a series of models to organize network data, including a two-layered data management model to organize the data in Internet of Things [8], an event-linked network (ELN) model for the organization of big sensing data [9], an approach to extract events and their internal links from large scale data leveraging predefined event schemas in the Web of Things [10] and a synergetic mechanism for digital library to provide information service in the mobile and cloud computing environment [11]. The differences between content-based and social network approach are: (1) the aim of content-based approach is to predict whether a specified number is a spam or not; (2) the social network approach always generates a directed graph from logs, then use algorithms to analyze the behavior of mobile users. Xu et al. used static, temporal and network features and the SVM and KNN algorithms to deal with the spammer classification problem [12]. However, it does not take temporal information into classification process. Balasubramaniyan et al. have proposed CallRank algorithm to deal with spam detection [13]. CallRank computes global reputation by Eigentrust algorithm in P2P networks [14]. However, the effectiveness of CallRank is unconvincing since the experiments are performed on the simulation data of VoIP system.

There are also some detection techniques to deal with Email spam which might facilitate the call spam detection. In [15], Ramachandran et al. used an algorithm based on eigencluster, which has been proposed by Cheng et al. [16] to detect Email spam. The algorithm has two phases: a divide phase and a merge phase. For the divide phase, the algorithm takes data with pair similarities as input, and then
uses an efficient implementation of the spectral algorithm to recursively partition data into two disjoint parts. However, the performance of the algorithm is not stable since the time and space requirement of the algorithm is dependent on the balancing of the partition process. Lam and Yeung [6] proposed 7 features for Email spam detection and then used the similarity weighted k-NN method to detect Email spam. The weights of the seven features are specified by user, and the effectiveness of the algorithm largely depends on user experience.

3. Problem Formulation

In this section, we present the problem formulation and define notations used throughout the paper.

3.1. Basic Concepts. In general, our study takes a call log file $F$ as input, where $F$ is composed of 4 fields: user, other, direction, and timestamp. The user field denotes the phone number of the specified user at observation, and the other field stands for the identification of the other user on the communication. The direction field represents the direction of the communication, incoming and outgoing. If it is incoming, the call is from other to user; otherwise, if it is outgoing, the call is from user to other. The timestamp field defines the time of communication. A call log file usually has another field named duration, which we do not take into consideration in our CLRank algorithm.

The original call log file $F$ can be transformed into a call log stream. Here, we present the definitions of call log stream and call log stream segment.

Definition 1 (call log stream). A call log stream $G$ is a series of directed graphs $G^t$, $G = (G^1, G^2, \ldots, G^T)$, which evolves infinitely over time. Each static directed graph $G^t$ represents the relationships of communications among $n^t$ nodes and $e^t$ edges. Here, $t \in [0, \ldots, T]$. In call log stream, node $v^t_i$ and $v^t_j$ denotes the person $i$ and $j$ with temporal information $t$, edge $e^t_{ij}$ denotes call information sent from $i$ to $j$ at time $t$. Here, $i, j \in [0, n^t - 1]$.

Definition 2 (call log stream segment). The series of static graphs between time interval $[t_s, t_{s+1} - 1]$ compose the $s$th call log stream segment $G^s$, $s \geq 0$. $G^s = (G^{s}, G^{s+1}, \ldots, G^{s+1-s-1})$.

The relations between two nodes $v^t_i$ and $v^t_j$ in the same call log stream segment can be classified as four types, since there may or may not be a communication event in either direction of each dyad.

Definition 3 (callee). Node $v^t_i$ is a callee of node $v^t_j$ if the edge $\langle v^t_j, v^t_i \rangle$ is contained in $G^t$.

According to the definition, Callees represents the relationship of the incoming links of a node. We denote the set $V^i_{i}^t$ as the incoming links of the node $v^t_i$.

Definition 4 (caller). Node $v^t_i$ is a caller of node $v^t_j$ if the edge $\langle v^t_j, v^t_i \rangle$ is contained in $G^t$.

Callers represent the relationship of the outgoing links of a node. We denote the set $V^O_{i}^t$ as the outgoing links of the node $v^t_i$.

Definition 5 (mutual friend). Node $v^t_i$ and node $v^t_j$ are mutual friends if $\langle v^t_i, v^t_j \rangle$ and $\langle v^t_j, v^t_i \rangle$ are contained in $G^t$.

Mutual friend represents the relationship of two users calling each other. Let $V^M_i = V^i_t \cap V^O_i$.

Definition 6 (stranger). Node $v^t_i$ and node $v^t_j$ are strangers if neither $\langle v^t_i, v^t_j \rangle$ nor $\langle v^t_j, v^t_i \rangle$ is contained in $G^t$.

Here, we propose our naïve solution of call spam detection. We propose two indexes named reputation and reciprocity to help to determine whether a node is a spammer. The definitions of reputation and reciprocity are motivated by the following observations.

1. A certified mobile user tends to have a lot of mutual friends, which means a user should make and also receive calls in a specified time interval. The call pattern of a certified user is usually bidirectional.

2. A spammer wants to quickly spread information to victims by communicating with many users intensively. These accounts have many outgoing links, while the number of incoming links is relatively small. The call pattern of a spammer is largely unidirectional. A spammer tends to have few mutual friends. There is a wide gap between the number of calls made and received by a spammer. Although call spam accounts have sent huge quantities of messages, only a small fraction of recipients will reply in a specified interval.

Definition 7 (reputation). The reputation of node $v^t_i$ is defined as the ratio between the number of callers and callees as

$$R_p(v^t_i) = \frac{V^I_{i}^t}{V^I_{i}^t + V^O_{i}^t + \psi}.$$  \hspace{1cm} (1)

Here, $\psi$ is a constant which is specified by user. If there is no parameter $\psi$ at the denominator, $V^I_{i}^t = 0$ and $V^O_{i}^t = 1$, according to the definition, the user is likely to be a spammer, which contradicts with the intuition.

From Definition 7 we can observe that if the numbers of the callees is much smaller than the amount of callers, the reputation is close to zero, $v^t_i$ is likely to be a spammer.
Definition 8 (reciprocity). The reciprocity of node \( v'_i \) is defined as

\[
R(\{v'_i\}) = \frac{\bigcup_{j} \bigcap_{i} v'_i}{\bigcup_{i} v'_i + \zeta}.
\]

(2)

Here, \( \zeta \) has the similar functionality as \( \psi \) in Definition 7. From Definition 8 we can observe that if few recipients send reply messages, then \( v'_i \) is likely to be a spammer.

3.2. PageRank and HITS Algorithms. PageRank computes a score for each web page by link analysis. It is motivated by the observation that a hyperlink from one web page to another is an authority transformation to the destination page. Given a page \( p \) with inlinks \( I_p \) and outlinks \( O_p \), the PageRank score \( PR \) is obtained by the following equation:

\[
PR_p = d \cdot \sum_{q \in I_p} \frac{PR_q}{|O_q|} + (1 - d) \cdot E_p.
\]

(3)

Here, \( d \) is the damping factor. \( E \) denotes the random page vector selected in the web graph. PageRank is simple, effective, and robust for malicious attacks. It is particularly suitable to rank a set of web pages or people.

Hubs and Authorities (HITS) filters search results for a specific topic [17]. Most of the web pages for searching a topic are of two types: authorities and hubs. The authority is a web page with good, authoritative content on a specific topic, and the hub is a web page pointing to many authoritative web pages. Given a topic, HITS first finds a subset of the web pages. The subset should have the characteristics of being small, consisting of pages relevant to the given query and containing most of the authoritative pages. The subset of the web pages is obtained by analyzing the connection of a root set of nodes. The root set contains the top \( k \) search results of the given query from a search engine. Then two scores, a hub score and an authority score are assigned to each node of the subset of web pages. The scores can be computed by an iterative algorithm.

4. Approach

In this section, we propose a two-stage framework and present the approach for each stage. The first stage uses a ranking algorithm CLRank; the second stage classifies the results. Temporal information is taken into consideration in both stages.

4.1. Preprocessing. The purpose of preprocessing is to generate mobile social network from call logs. First, the input of call log stream is partitioned into many call log stream segments. The logs can be represented as a \( m \times n \times k \) data cube \( D \). Here, \( m \) denotes the number of mobile nodes that make phone calls to any of \( n \) destinations within \( k \) time intervals. Thus, \( D_{ij}^t \) represents the times of node \( v'_i \) calling \( v'_j \) in interval \( t \).

The second step of preprocessing collapses the time axis to obtain a \( m \times n \) matrix \( M' \):

\[
M' = \sum_{t=0}^{k-1} D_{ij}^t.
\]

(4)

The strategy of assigning different weights to different edges by timestamp may improve the performance of our algorithm. Here, we omit this step for simplicity. The length of the call log stream segment is 5 days in our settings.

4.2. CLRank. The goal of CLRank is to assign a rank to each mobile node in call log stream segment and to use this value (1) to decide whether each caller is a spam or not and (2) to rank nonspam nodes.

CLRank is composed of two steps:

1. Select a seed set of nodes with high reputation in the call log social network;
2. Apply CLRank algorithm to the call log graph to compute the final score based on the seed set in procedure 1.

Since the final CLRank scores are based on the seed set in step 1, it is very important for seed node selection in procedure 1 to exclude spammers. In reality, there are three methods to obtain the seed set. A manual method which requires the selection process is processed by human. However, it is very time-consuming and human tend to make mistakes. An automatic method means to use seed selection algorithms to obtain the seed set. This method has the advantage of high efficiency. However, the effectiveness of selection algorithm is critical. A semiautomatic method is composed of two steps: first, the algorithm uses an automatic method to obtain an intermediate result; then the final seed set is determined by human. In this paper, we use a semiautomatic method to compute the seed set.

Suppose the whole rank value in the given call log graph is \( \omega \), our seed selection algorithm adds the highest rank of the nodes \( R_h \) in call log graph such that the rank of \( R_h \) equals 10 percent of \( \omega \). We set the size of this intermediate result as the minimum of \( R_h \) and one percent of the total number of mobile nodes in the call log graph. Since call log conforms to power-law distribution [18], the intermediate result excludes most of the spammers effectively. We then examine the intermediate result to obtain final seed set \( \lambda \).

The CLRank algorithm is presented in Algorithm 1.

CLRank computes vector \( s' \) representing communication pattern for each mobile node, and uses a threshold \( \tau \) to classify nodes in call log graphs. Given the behavior pattern of a mobile node, CLRank returns a score, which can classify the node as:

1. A spammer node, if the score is larger than \( \tau \).
2. A normal node, if the score is less than \( \tau \).
3. An undefined node, if the node is not in the call log graph.

Since the behavior patterns of spammer might vary from time to time, CLRank should recalculate \( s' \) accordingly.
Input:
- $t$: a specified time interval
- $M^t$: an $m \times n$ call log graph in $t$
- $N^t$: number of nodes in $t$
- $\lambda$: seed set $\lambda$ obtained by semi-automatic method
- $\alpha$: decay factor. $\alpha = 0.85$
- $\epsilon$: precision threshold

Output:
- $s^*$: CLRank vector of scores

Variables:
- $T^t$: Markov chain transition probability matrix
- $s$: vector of scores
- $\sigma$: error between $s$ and $s^*$

Algorithm:
1. For Each node $v_i^t$ in $M^t$
2. If $\langle v_i^t, v_j^t \rangle \in M^t$
3. For Each connection from $v_i^t$ to $v_j^t$
   - $T_{v_j^t, v_i^t}^t = 1/N_i$
4. Else For Each node $v_i^t$ in $M^t$
   - $T_{v_i^t, v_j^t}^t = 1/N^t$
5. Let $T^t = \alpha T^t + (1 - \alpha)E$, with
   - $E[v_i^t] = [1/||\lambda||]_{N^t \times 1}$, if $v_i^t \in \lambda$;
   - $E[v_i^t] = [0]_{N^t \times 1}$, otherwise
6. Initialize $s = [1/N^t]_{N^t \times 1}$ and $\sigma = \infty$
7. While $\sigma < \epsilon$
   - $s^* = T^t \cdot s$
   - $\sigma = \|s^* - s\|$
8. return $s^*$

Algorithm 1: CLRank algorithm.

However, it is difficult to determine the time of recalculation. CLRank is based on PageRank algorithms; the simple way is to recompute $s^*$ at a fixed interval. In practice, CLRank performs recalculation if the behavior of the nodes cannot match any known patterns.

4.3. CLRank System Architecture. CLRank can be deployed on the server of mobile network operator. The server uses call logs from all mobile users to perform CLRank algorithm and utilizes the results to detect spammers. The results of CLRank algorithm are assigned with a time to live field to record its lifetime. After the specified lifetime, the server runs the CLRank algorithm again so it can detect nodes which behave normal for some time, and after that, it performs malicious behavior. If a phone number is recognized as a spammer, the system blocks the communication of the phone number for some time. The system might also send voice prompt to the user to explain the reasons of blockage. There is no need to install software on mobile phone; thus, our algorithm can provide protection for smartphones as well as feature phones.

5. Experimental Evaluation

In this section, we evaluate the effectiveness of CLRank algorithm. We describe experimental data and environment in Section 5.1. We present results for our real dataset in Section 5.2. We briefly discuss the effects of parameter values in Section 5.3.

### Table I: A sample record from city C.

| User | 158 * * * * * * * *
| Other | 138 * * * * * * * *
| Direction | Incoming
| Timestamp | 08:11

Section 5.1. Experimental Setting. We use real mobile call data set to perform experimental evaluation. Our data set is composed of call logs from city C (The name of the city is not provided as requested by the data provider) collected by a mobile network operator in China. The dataset covers over 10 million people and includes mobile call records of 30 days. It contains rich information, such as related cell tower during phone call. The positional accuracy of cell tower data is within 300–500 m. Unlike MIT Reality dataset or Nokia mobile dataset, our dataset does not provide GPS or WLAN data. We extract 4 fields from elementary data for experiments: user, other, direction, and timestamp. Table 1 presents a sample record. The information has been preprocessed for privacy issues.

CLRank could compute ranking scores of nodes which reflect the sending patterns in a specified interval. We build a
call spam classifier by constructing $M'$ from the call logs. We use the call log graph for a time interval of 5 days to construct CLRank classifier, so there are totally 6 time intervals in our experiments, denoted as $K_1 \sim K_6$. The threshold $\tau$ was set to 0. We compare the results of CLRank classifier with victim spam reports by mobile applications, for example, Tencent Mobile Manager and Qihoo 360 Mobile Security Solution. The victim spam reports are obtained manually. We have installed these two applications on smartphones and manually dial each phone number. The software prompts that whether this phone number has been marked as spammers, and how many times it has been reported by other people. We do not check the spams which have not been detected by CLRank, since it is very difficult to perform this task manually. We conduct all the experiments on a Pentium-4 3.0 GHz PC with 4 GBytes of main memory, running on CentOS 4.5 operating system. We implement our algorithm in C++ using GCC 4.5.4 on GraphChi 0.2.1.

5.2. Accuracy Evaluation. The accuracy evaluation results of 6 time intervals are displayed in Figure 1.

Figure 1 shows the correctly classified and misclassified spam numbers by CLRank. The accuracy of CLRank is 84.5–91.8%. The results of CLRank have been affected by the sparsity of $M'$. Some of the phone numbers only receive calls, and they do not initiate phone call activities. This makes some normal users in $M'$ be taken as spammers, since they have been assigned low ranking scores. We observe that more than 300 spammers are identical in the 6 time intervals. This result implies that the number of spammers remains relatively stable throughout our data collection process, and it implies the effectiveness of our algorithm. We also observe that the behavioral patterns of some smartphone numbers change radically throughout the 6 time intervals. These smartphones might have been utilized to perform spam activities or have been infected by malware which the users are unknown.

It is very difficult for us to determine the values of reputation and reciprocity while analyzing call spam by our naive solution. We have tried different parameter values and selected the parameters which lead to high confirmed and low misclassified values. Figure 2 shows the results by our Naïve solutions with the optimal parameter values ($R_p(v^c) = 0.07$, $R_c(v^c) = 0.09$, $\psi = \zeta = 50$). The result indicates that the naïve solution has poor accuracy. The correctly classified nodes are lower than CLRank while the misclassified spammers are higher.

5.3. CLRank versus Other Ranking Algorithms. PageRank is a classical algorithm for web spam detection. Figure 3 shows the correctly classified and misclassified spam numbers on the same dataset by PageRank. The accuracy of PageRank is 79.1–82.6%. The number of correctly classified calls is lower than CLRank while the number of misclassified calls is higher and thus results in lower detection rate. The reason of this phenomenon is that the PageRank algorithm does not have high reputation node selection process. It is possible that some disguised spammers receive high PageRank scores.

Now we introduce the concept of demotion [19]. Here, it denotes the set of nodes which receive high scores from PageRank while receive low scores from CLRank. The demotion reflects the effectiveness of CLRank to cut down the influence of spams. Figure 4 shows the demotion in CLRank. The vertical axis of Figure 4 shows the number of nodes which receive high scores from PageRank got demoted in TrustRank. Blue bars represent spam. Yellow bars denote the reputable nodes. We point out that CLRank detects most of the spams from top-scored users effectively, and the result of reputable ones detected by CLRank is quite stable. CLRank ensures that top-scored users are reputable users.

CallRank is a social network ranking algorithm for combating spam over internet telephony by call duration and global reputation. Figure 5 shows the number of correctly classified and misclassified spammers on the same dataset.
by CallRank. The accuracy of CallRank is 83.3~89.4%. The results are better than PageRank, and are better than CLRank in \( K_4 \) and \( K_5 \). The reason of this phenomenon is that the CLRank algorithm has taken the duration of call into consideration.

### 5.4. Effects of Parameter Values

We have tested the effects of parameter values on the results of CLRank. If we set parameter \( \tau \) to 0, our algorithm is able to compute the transitive closure of call log graph from the seed set, thus it can detect most of the spammers effectively. If we use a larger parameter \( \tau \), it is possible for our algorithm to detect the spammers if some normal nodes vote for them. However, some normal nodes with low ranking scores might also be taken as spammers.

### 6. Discussion and Conclusions

#### 6.1. Discussion

We discuss some possible extensions of our CLRank algorithm.

1. **Accuracy.** We can assign different weights to different edges when collapsing time axis during the second step of preprocessing. The weight can be determined by timestamp and the duration of the call. This extension will improve the accuracy of our algorithm.

2. **Efficiency.** We can design and implement our solution by parallel technology. A number of platforms, for example, Apache Hadoop, have been developed for this purpose in the context of big data. We are applying these techniques to mobile data and implementing faster algorithms.

#### 6.2. Conclusion

In this paper, we have proposed a server-based solution for detecting spammers automatically. Our algorithm takes a call log file as an example, to analyze the characteristics of call logs and discover the potential call spammers. A time-based graph model and a simple and effective algorithm CLRank with ranking and classification were proposed to find potential spammers. The detection process of CLRank can be performed on the server of mobile network operator, without any involvement of users. To show the effectiveness of CLRank, we have performed experiments on a real data set provided by a Chinese mobile network operator. Experimental results show that our proposed model can find spammers from call logs effectively (with 84.5~91.8% of accuracy) without any manual interventions.

### Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.
Acknowledgments

This research is sponsored by The National Natural Science Foundation of China Civil Aviation joint Fund (Grant no. U1433116), the Fundamental Research Funds for the Central Universities (NZ2013306), the 333 Project of Jiangsu Province and the Technology Foundation (JSJC2013605C009).

References

[1] N. Jabeur, S. Zeadally, and B. Sayed, “Mobile social networking applications,” Communications of the ACM, vol. 56, no. 3, pp. 71–79, 2013.

[2] A. Chin and D. Zhang, Mobile Social Networking, Springer, New York, NY, USA, 2014.

[3] A. Narayan and P. Saxena, “The curse of 140 characters: evaluating the efficacy of SMS spam detection on Android,” in Proceedings of the 3rd ACM Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM ’13), pp. 33–42, ACM, November 2013.

[4] K. Yadav, P. Kumaraguru, A. Goyal, A. Gupta, and V. Naik, “SMSAssassin: crowdsourcing driven mobile-based system for SMS spam filtering,” in Proceedings of the 12th Workshop on Mobile Computing Systems and Applications (HotMobile ’11), pp. 1–6, March 2011.

[5] P. O. Boykin and V. P. Roychowdhury, “Leveraging social networks to fight spam,” Computer, vol. 38, no. 4, pp. 61–68, 2005.

[6] H.-Y. Lam and D.-Y. Yeung, A Learning Approach to Spam Detection Based on Social Networks, Hong Kong University of Science and Technology, 2007.

[7] C.-Y. Tseng and M.-S. Chen, “Incremental SVM model for spam detection on dynamic email social networks,” in Proceedings of the IEEE International Conference on Social Computing (SocialCom ’09), pp. 128–135, Vancouver, Canada, August 2009.

[8] Y. Sun and A. J. Jara, “An extensible and active semantic model of information organizing for the internet of things,” Personal and Ubiquitous Computing, 2014.

[9] Y. Sun, H. Yan, J. Zhang et al., “Organizing and querying the big sensing data with event-linked network in the internet of things,” International Journal of Distributed Sensor Networks, vol. 2014, Article ID 218521, 11 pages, 2014.

[10] Y. Sun, H. Yan, C. Lu, R. Bie, and Z. Zhou, “Constructing the web of events from raw data in the web of things,” Mobile Information Systems, vol. 10, no. 1, pp. 105–125, 2014.

[11] J. Zhang, Y. Sun, L. Zhu, and X. Qiao, “A synergistic mechanism for digital library service in mobile and cloud computing environment,” Personal and Ubiquitous Computing, 2014.

[12] Q. Xu, E. W. Xiang, Q. Yang, J. Du, and I. Zhong, "SMS spam detection using noncontent features," IEEE Intelligent Systems, vol. 27, no. 6, pp. 44–51, 2012.

[13] V. Balasubramaniyan, M. Ahamad, and H. Park, “CallRank: combating SPIT using call duration, social networks and global reputation,” in Proceedings of the 4th Conference on email and anti-spam (CEAS ’07), August 2007.

[14] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, “The eigentrust algorithm for reputation management in P2P networks,” in Proceedings of the 12th International Conference on World Wide Web (WWW ’03), pp. 640–651, ACM, May 2003.

[15] A. Ramachandran, N. Feamster, and S. Vempala, “Filtering spam with behavioral blacklisting,” in Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS ’07), pp. 342–351, ACM, November 2007.

[16] D. Cheng, R. Kannan, S. Vempala, and G. Wang, “A divide-and-merge methodology for clustering,” ACM Transactions on Database Systems, vol. 31, no. 4, pp. 1499–1525, 2006.

[17] J. M. Kleinberg, “Authoritative sources in a hyperlinked environment,” Journal of the ACM, vol. 46, no. 5, pp. 604–632, 1999.

[18] L. A. Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Huberman, “Search in power-law networks,” Physical Review E, vol. 64, no. 4, Article ID 046135, 2001.

[19] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen, “Combating web spam with trustrank,” in Proceedings of the 30th International Conference on Very Large Data Bases, vol. 30, pp. 576–587, VLDB Endowment, 2004.