

Research Article

CPSFS: A Credible Personalized Spam Filtering Scheme by Crowdsourcing

Xin Liu,¹ Pingjun Zou,¹ Weishan Zhang,¹ Jiehan Zhou,² Changying Dai,¹
Feng Wang,¹ and Xiaomiao Zhang¹

¹College of Computer & Communication Engineering China University of Petroleum (East China), Qingdao, China

²University of Oulu, Oulu, Finland

Correspondence should be addressed to Xin Liu; lx@upc.edu.cn

Received 12 September 2017; Accepted 5 December 2017; Published 27 December 2017

Academic Editor: Kuan Zhang

Copyright © 2017 Xin Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Email spam consumes a lot of network resources and threatens many systems because of its unwanted or malicious content. Most existing spam filters only target complete-spam but ignore semispam. This paper proposes a novel and comprehensive CPSFS scheme: Credible Personalized Spam Filtering Scheme, which classifies spam into two categories: complete-spam and semispam, and targets filtering both kinds of spam. Complete-spam is always spam for all users; semispam is an email identified as spam by some users and as regular email by other users. Most existing spam filters target complete-spam but ignore semispam. In CPSFS, Bayesian filtering is deployed at email servers to identify complete-spam, while semispam is identified at client side by crowdsourcing. An email user client can distinguish junk from legitimate emails according to spam reports from credible contacts with the similar interests. Social trust and interest similarity between users and their contacts are calculated so that spam reports are more accurately targeted to similar users. The experimental results show that the proposed CPSFS can improve the accuracy rate of distinguishing spam from legitimate emails compared with that of Bayesian filter alone.

1. Introduction

Email is an essential communication method in the Internet age. However, the abuse of bulk emails allows spam to spread like a plague. Spam consumes network bandwidth and brings also other threats to recipients: unwanted advertisements and pornographic content, as well as malicious viruses [1]. A spammer does not need to get permissions from recipients when sending spam, which causes serious annoyance to people and even leads to information security risks [2]. If a recipient clicks a malicious link in the spam message, their personal information may be automatically sent to the spammer via a malicious program, which is an obvious challenge for privacy protection [3, 4]. Statistics showed that spam accounted for 81.8% of total emails in 2016, compared with 72.9% in 2015 [5], which is obviously an increasing threat to email users.

To tackle this issue, in this paper, we classify spam into two categories according to the scope of affected users. One kind of spam is “complete-spam,” which is defined as email

identified by all users as spam. The other kind of spam is semispam, which is identified as spam by some users but as legitimate by other users. Most spam filters were developed to identify complete-spam [6, 7]. The accuracy of spam detection of some of these filters can be fairly high [8]. However, very often we still find spam in our email inbox. This is because the existing spam filters can only identify complete-spam, but not semispam.

To resolve this issue, we need a comprehensive and personalized filtering mechanism that can utilize user contacts to collaboratively identify semispam, which is a novel scheme called CPSFS—Credible Personalized Spam Filtering Scheme. In CPSFS, a user can make use of social networks to obtain spam reports from his or her contacts, which can then be used to filter both complete-spam and semispam. This is an approach using crowdsourcing from involved email users, where spam reports from credible similar contacts help to boost the performance of collaborative spam filtering.

But how to choose credible contacts for a user? Users on the Internet are interconnected to form social networks

```

Input:
  L: local spam list
  M: an email set from inbox
  T: local trust-similarity list
Output:
  L: updated spam list
(1) Let  $M_i$  is the  $i$ th email of  $M$ ,  $L_j$  is the  $j$ th item of  $L$ ,  $T_k$  is  $k$ th contact's trust,  $S_k$ 
    is  $k$ th contact's similarity,  $T_{M_i}$  is the trust of the sender of  $M_i$ ,  $n$  is the number of
    emails,  $m$  is the number of items in  $L$ ,  $c$  is the number of contacts,  $S_r$  is the
    generated spam report,  $T_h$ : trust threshold,  $S_h$ : interest similarity threshold, flag is
    the subject of email
(2) for  $i \leftarrow 1$  to  $n$  do
(4)   if flag = "spam report"
        if  $T_{M_i} \geq T_h$ 
             $L_{m+1} \leftarrow$  content of the email
             $m \leftarrow m + 1$ 
            for  $k \leftarrow 1$  to  $c$  do
(8)               if  $S_k \geq S_h$ 
                    forwarding  $M_i$  to contact  $k$ 
                end if
(10)            end for
            end if
        else
             $MD_5 \leftarrow$  hash( $S_i$ )// Calculating MD5 hash for  $S_i$ 
            for  $j \leftarrow 1$  to  $m$  do
                if  $MD_5 = L_j$ 
                    put  $S_i$  into junk box
                end if
            end for
             $L_j \leftarrow MD_5$ 
             $m \leftarrow m + 1$ 
(6)            generating the spam report  $S_r$  of  $S_i$ 
(7)            for  $k \leftarrow 1$  to  $c$  do
(8)                if  $S_k \geq S_h$ 
                    sending  $S_r$  to contact  $k$ 
                end if
(10)           end for
(11) end for
(13) return  $L$ 

```

ALGORITHM 1: Local spam filtering algorithm.

according to their relationships [9–11]. Social trust is a key factor that affects the sharing of knowledge and the development of social relationships [12, 13]: users are more likely to accept suggestions from others with high trust value and interests similarity [14]. Social trust can be calculated by analyzing Social Computing [15, 16].

Our contributions in this paper are as follows.

(1) We classify spam into two categories, complete-spam and semispam. We design different methods for filtering these two kinds of spam at both email servers and clients.

(2) We propose CPSFS which uses a crowdsourcing mechanism to filter semispam, where users with similar opinions collaborate together against semispam by sharing spam reports with each other, and social trust is used for users to choose credible contacts in order to avoid malicious users exploiting our scheme to propagate spam.

This paper is organized as follows. Section 2 discusses related work. Section 3 explains how the CPSFS is deployed.

Section 4 describes how to calculate trust value and interest similarity. The underlying local filter algorithm is presented in Algorithm 1. Next, Section 6 evaluates our approach via the experimental validation and compares with other filters. Finally, Section 7 concludes our work.

2. Related Work

Most previous works tried to filter out complete-spams for all users. We divided the existing work into four types based on the used techniques: the Black/White List, Bayesian, Machine Learning, and Social Computing.

2.1. Black/White List. Jaeyeon and Emil [17] presented a black/white list approach that relies on the number of IP addresses to determine whether an email is spam. The black list includes an email server and an IP address of sender. If the source of email appears in the black list, the email is

identified as spam. The problem with a black/white list is that it is difficult to update and maintain the list.

2.2. Bayesian Approach. O'Brien and Vogel [18] applied the Bayesian algorithm for spam filtering. The Bayesian filter parsed emails into keywords and then computed probabilities of keywords that appeared in spam and legitimate emails, respectively. The results showed that the Bayesian filter detected 91.7% of spam. This is a relatively high recognition rate for all emails using uniform criteria without considering semispam emails while calculating recognition rate.

2.3. Machine Learning Approach. Haider et al. [19] presented a machine learning-based approach by detecting batches of emails to filter spam effectively. The filter needs to be trained to distinguish keywords in spam. Scholkopf and Platt [20] presented a method that minimizes a loss function with respect to user's personal distribution based on the available biased samples. However, it is difficult to make the sample data have the same Dirichlet distribution.

2.4. Social Computing Approach. Zisiadis et al. [21] presented a collaborative method for email filtering called Mailbook which was based on a social network. Each node could mark the received spam and stored it in its own database. If one node marked an email as spam, the votes of spam increased by 1. Once the votes of spam reached a certain number, the system would mark the email as spam automatically. Similarly, Boykin and Roychowdhury [22] proposed a spam filtering approach based on social networks, which allows users to share the spam information with their friends to identify spam.

Sirivianos et al. [23] applied social network and trust mechanism for spam filtering. A node in a social network may report a spammer's IP address to a centralized server for the spam it received. The centralized server calculated the trust value according to the degree of confidence and credibility of the spam reporter; then it decided whether the IP address is a spammer. This method required a central server, which increased additional network overhead, and its accuracy was reduced by dynamic IP addressing. Shen and Li [24] presented a social network-aided spam filter which is used to improve the accuracy of spam filtering by integrating four new components into a Bayesian filter; these components identify spam by the closeness of nodes. Each node needs to collect information and check spam by its social network-aided spam filter, which will increase the overhead of the system.

Apparently, if users can share information on spam with their friends with similar interests, they can help each other to identify spam emails more accurately.

3. Designing Spam Filtering Scheme

Users and their contacts with the same interests are called "similar contacts." We assume that similar contacts always have the same opinion on the same email; then we can design

TABLE 1: Local spam list.

ID	MD5	Contact
(1)	469352d907cb67bc2b228e8b0a839eee	Zhang_ch@163.com
(2)	daeb67d732741a4982d6929ee191e210	jamesell@163.com
(3)	6d49148666475138cec9f42cc29a7cd7	qingzhi@163.com
(4)	bac0b74229c3f73757fe72508e25471a	hannan@gmail.com
(5)	2330ead823cd690611b9b990e29cc283	yangxf@upc.edu.cn

a scheme in which similar contacts share their information on spam with each other to filter semispams.

Our scheme consists of two modules: (1) Bayesian-based spam filtering deployed at an email server for all users; (2) credible similarity-based spam filtering by crowdsourcing deployed at each user local host. The structure of the proposed CPSFS is showed in Figure 1.

We deploy the Bayesian spam filtering at email servers to filter complete-spam for all users because the more the emails used for training are, the more accurately the Bayesian filter identifies spam.

The spam filtering deployed at local system uses three lists for storing information on contacts and spam reports from other users, namely, local trust list, local spam list, and local interest list. A user gets his contacts' interests and disinterests by exchanging interest lists to calculate their similarity between them. Users share information on spam with their credible contacts by pushing spam reports to their contacts. Spam reports generated at the local host and that from contacts are stored in the local spam list. At the local system, before a user browses his inbox, those emails in the inbox are checked according to spam information from the local spam list. A credible contact means the similarity and trust between the user and the contact are higher than the corresponding threshold. We construct an interest list and a trust-similarity list at each local host in order to calculate the similarities and social trust values.

3.1. Local Spam List. A local spam list contains an MD5 hash of the spam from spam reports and the email address which the spam report is from, as shown in Table 1.

A spam report consists of a MD5 hash of the corresponding spam, which is obtained from the content of spam email to avoid spam with an altered subject heading or a forged address. Some spam reports were generated automatically at local system. Others were from users' credible contacts.

A user may receive different spam reports on the same email from different contacts. Of these, only the report from contact with the highest interest similarity will be recorded into the local spam list; other reports will be dropped.

3.2. Local Interest List. The interests of a user in a social network represent the user's personality [25]. There are similar interests among users. The common disinterests between two users also indicate whether they are similar in some way. Therefore we encourage each email user to fill their own interests and disinterests in their local interest lists and exchange their lists with each other via emails.

TABLE 2: Local Interest List.

ID	Interest	Disinterest
(1)	Shopping, movie, music, food, car	Pet, beauty, drawing, cartoon, IT
(2)	Pet, shopping, food, car, reading	Game, music, movie, basketball, IT
(3)	Car, shopping, music, food, game	Singing, beauty, cartoon, drawing, reading
(4)	Music, food, movie, beauty, drawing	Shopping, music, pet, cartoon
(5)	Car, pet, cartoon, music, food	Shopping, movie, basketball, IT, beauty
(6)	IT, movie, music, basketball, reading	Food, car, singing, game, dancing
(7)	Car, food, basketball, game, IT	Movie, shopping, singing, dancing, game

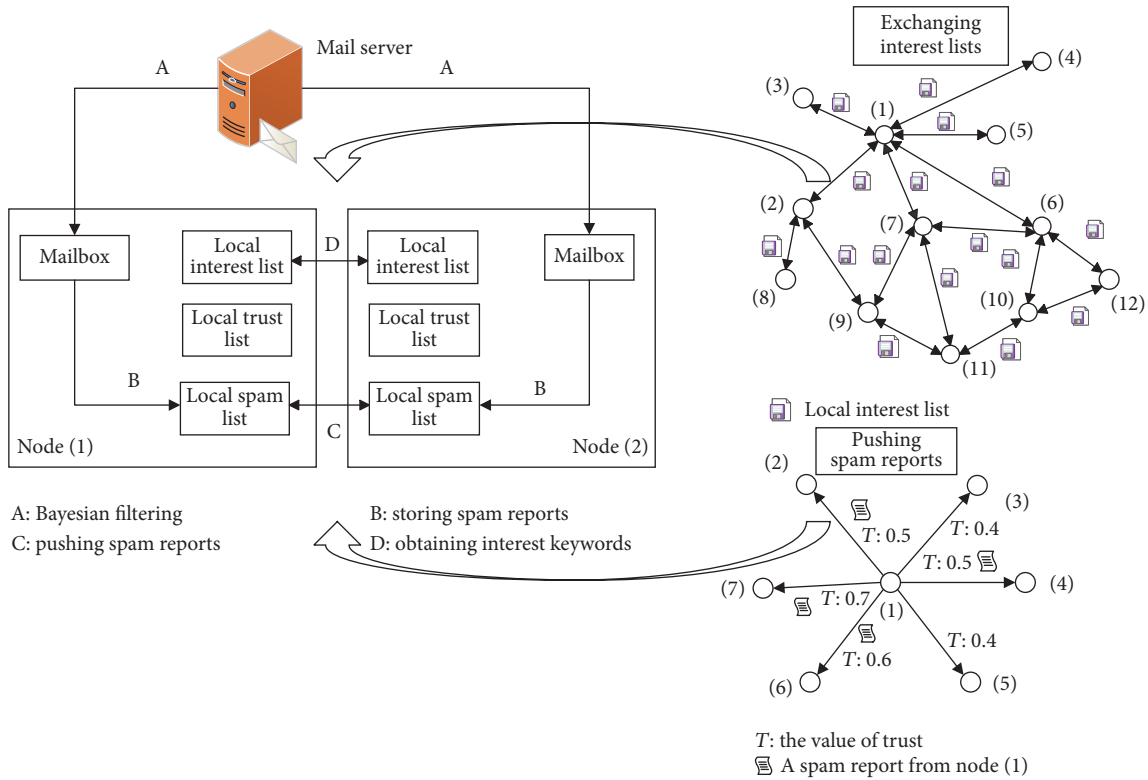


FIGURE 1: The structure of the proposed CPSFS.

These interests and disinterests can be described by some keywords. The interests and disinterests of a user and the user's contacts are stored in the local interest list as shown in Table 2. Each contact of the user has an ID in the local system. We can get the corresponding trust value and the interest similarity according to the contact's ID. Once a user gets a spam report, the local system will check the local trust-similarity list to push the spam report on this email to those credible contacts.

3.3. Local Trust-Similarity List. Social trust is a measure of credibility in social networks. It reflects a certain degree of similarity between users, such as likes, dislikes, social relationships, and the closeness of their interests. Direct trust is calculated initially from the historical record of direct contacts between users which indicates the direct friendships (not including friends of a friend) between users and their contacts; then it can be adjusted according to the similarities between them.

TABLE 3: Local trust-similarity list.

ID	Email address	Trust value	Interest similarity
(1)	Zhang_ch@163.com	1.00	1.00
(2)	jamesell@163.com	0.52	0.25
(3)	qingzhi@163.com	0.79	0.42
(4)	hannan@gmail.com	0.56	0.33
(5)	yangxf@upc.edu.cn	0.41	0.33
(6)	dswang@upc.edu.cn	0.60	0.17
(7)	jiaozzy@gmail.com	0.73	0.11

The interest similarity is calculated according to interests and disinterests between a sender and its recipient, which indicates the closeness between them.

Trust values and similarities of contacts on a local host are stored in a local trust-similarity list, as shown in Table 3.

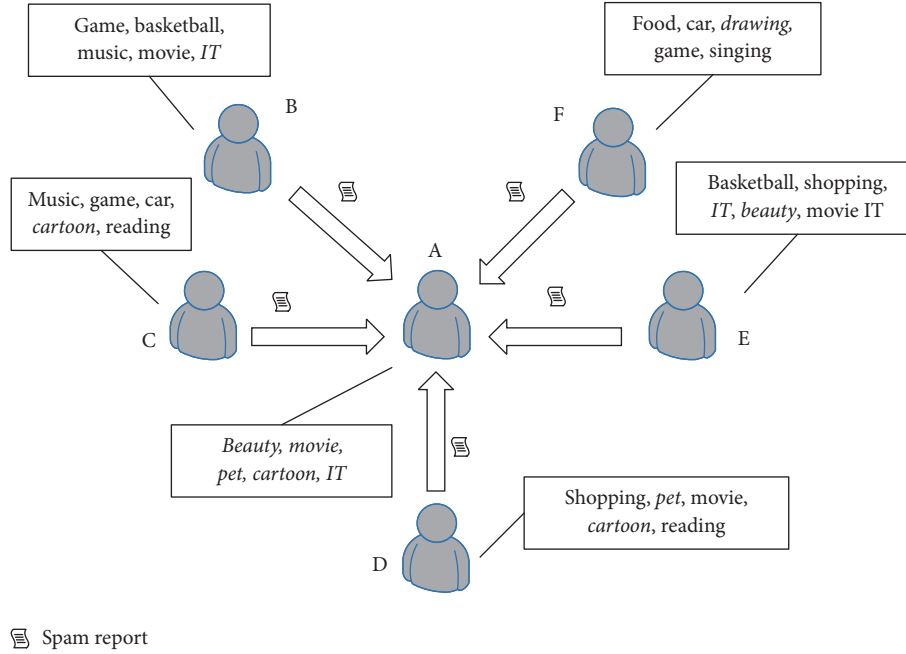


FIGURE 2: A simple example for obtaining personalized spam reports.

The email users and their relationships between them formed an email network. A simple example is shown in Figure 2, where nodes represent users and links represent the relationships between the user and their contacts.

3.4. Obtaining Personalized Spam Reports at a Client. A CPSFS client can filter semispam automatically by obtaining personalized reports from a user's contacts. When a user node and his/her contacts have more similar interests and disinterests, the user is more likely to obtain personalized spam reports from his or her contacts. A simple example is shown in Figure 2. We list the disinterests for each user in this figure. User A gets spam reports about *IT* from B, spam reports about *cartoon* from C, spam reports about *pet* and *cartoon* from D, spam reports about *IT* and *beauty* from E, and spam reports about *drawing* from F.

4. Calculating Trust Value and Interest Similarity

4.1. Calculation of Interest Similarity. The more the mutual interests and disinterests between a user and his or her contacts are, the more similar they are [26]. We calculate the similarity between node A and node B via

$$S(A, B) = \frac{(M_{00} + M_{11})}{(M_A + M_B - M_{11} - M_{00})}, \quad (1)$$

where $S(A, B)$ is the similarity between A and B. M_A represents the number of A's interests and disinterests. M_B represents the number of B's interests and disinterests. M_{00} represents the number of the mutual disinterests. M_{11} represents the number of the mutual interests.

4.2. Calculation of Trust Value. We use the additive increase/multiplicative-decrease algorithm [27] to adjust the trust value between nodes. In this algorithm, initially, the trust value of users' contacts should be assigned. Alternatively, the trust value of a contact can be calculated according to the number of emails from the contact. We count each contact's emails in the inbox for the past month. The number of emails from the i th contact is represented by num_i . Let max represent the largest number of emails. The initial trust value of a contact is $T_i = num_i / max$.

If the similarity between nodes is changed, the trust value will be changed correspondingly. We use b to indicate the degree of trust value changes. So the trust value will be changed via

$$t_{ij} = t_{ij} \pm b \quad (0 < b < 1). \quad (2)$$

If the interest similarity is higher than the threshold of similarity and the trust value between them is less than the given trust threshold, the trust will increase b in formula (2). Otherwise the trust value will decrease b . It is important to set an appropriate value for b . We will discuss how to adjust the trust value threshold rationally in the evaluation section.

5. Spam Filtering Process at Local System

Users can mark an email manually or automatically as spam at local system.

5.1. Mark an Email Automatically. All emails of a user are examined by a Bayesian filter at an email server before they reach clients [28]. When the user logs in, the local system should check all emails in the inbox. If there is a spam report in the inbox, the content should be extracted and written

into the local spam list and then the system should forward the email to similar contacts whose similarity is beyond the threshold. Otherwise, the system will calculate the MD5 hash for this email, if an email is identified as spam using the local spam list. That is to say, when the MD5 hash matches an item in the local spam list, the spam is put into a junk box and the spam report is pushed to the similar contacts. The corresponding filtering process is handled by *Algorithm 1 LocSpamFilter*.

5.2. Mark an Email Manually. While a user browses the inbox, he or she may find some spam. Once the user puts a spam email into a junk box, the local system will generate a spam report, add it to the local spam list, and push it to similar contacts.

Pushing a spam report is accomplished via sending emails [29]. The subject of the email is *spam report*, which is used to distinguish spam report from other emails. The content of this kind of email is the MD5 hash of this spam. When a user identifies an email as spam, the spam report is automatically generated and pushed to similar contacts.

The trust value is used to limit the recipients of spam reports from the credible users. If the trust value of a contact is over a given threshold, this user will receive spam reports from this contact. If the similarity to a contact is above a given threshold, the local system will push spam reports to the contact. The similarity threshold is used to limit the scope of spam report propagation to similar contacts only. This will both improve the accuracy of identifying semispam, and reduce network overhead. We will discuss the threshold of trust and interest similarity in the evaluation section.

6. Experiments and Evaluations

6.1. Simulation Settings. The social network we used in our experiments is from Datatang and contains 1133 nodes and 10903 edges [30]. The average number of contacts of each node is 9.63. In addition, we use a sample set of 3000 emails including 1000 spam emails and 2000 legitimate emails from SpamAssassin public mail corpus [31]. We choose 1000 emails from the sample randomly as definite sample, and 500 emails from email boxes of our researchers as indefinite samples. We set the total number of interest keywords to 15, and the average number of interest keywords for each user to 10; then we sent an email of the 1500 emails randomly to 10% of the total nodes.

In our experiments, for formula (2), the initial trust value t_{ij} is randomly set from 0.5 to 1.0, the threshold of trust value is 0.5 and the initial value of b is 0.1. If a node received a spam email report from his friend and the trust value between them is less than the given trust threshold, the trust value between him and his friend will increase by b , which can make a user become trusted by his friend.

6.2. Comparisons with Other Methods and Accuracy of the CPSFS. We calculate the accuracy of the CPSFS by comparing with the Bayesian filter. The accuracy rate of filtration is R_a . We calculate the accuracy rate via

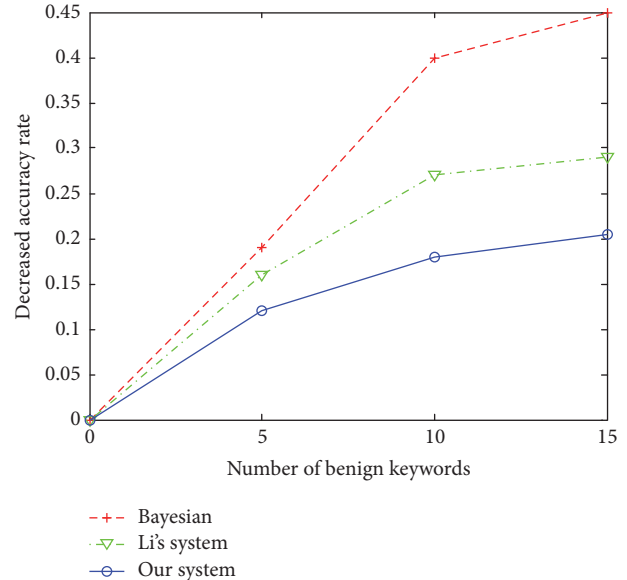


FIGURE 3: The accuracy rate under poison attacks.

$$R_a = \frac{n_1 + (N - n_1) \times \left(\frac{n_f}{n_i}\right) \times \left(\frac{n_c}{n_s}\right)}{N}, \quad (3)$$

where N is the number of emails used in our experiments. The number of email correctly classified by Bayesian filter is denoted by n_1 . n_i is the number of emails classified incorrectly by Bayesian filter. n_f is the number of emails classified as legitimate but are actually spam. When an email is classified correctly by the CPSFS, two cases are supposed to be considered: (1) a user considers it as a legitimate email, and the CPSFS did not mark it as spam; (2) a user considers it as spam, and the CPSFS marked it as spam. If an email is classified by the system correctly, we will record the number of users, which is denoted as n_c . n_s is the total number of the users in the social network. We evaluate the system using the same settings as in Section 6.1. The results from formula (3) show that our CPSFS has a higher accuracy rate than that of the Bayesian filter and Li's work (95.1% versus 91.4% versus 93.9%).

Poison attack is that an attacker adds benign keywords into emails intentionally in order to avoid spam to be identified by a spam filter. As word segmentation is the basis of a Bayesian filter, keywords of mail contents have a crucial impact on classification and performance of a Bayesian filter. The decreased accuracy is the decreased value of accuracy when the poison attacks happen. Figure 3 shows the decreased accuracy of the Bayesian filter, Li and Shen's work [27], and the CPSFS when they are subjected to poison attacks. When the number of benign keywords is set to 0, 5, 10, and 15, the accuracy of our CPSFS decrease smoothly. But the accuracy of the Bayesian filter decreases more quickly than that of our CPSFS because the Bayesian filter is entirely dependent on the detection of spam content. Our CPSFS reduces the effects of poison attacks on accuracy by sharing information on spam using crowdsourcing. The CPSFS considers the association of an email and its recipients,

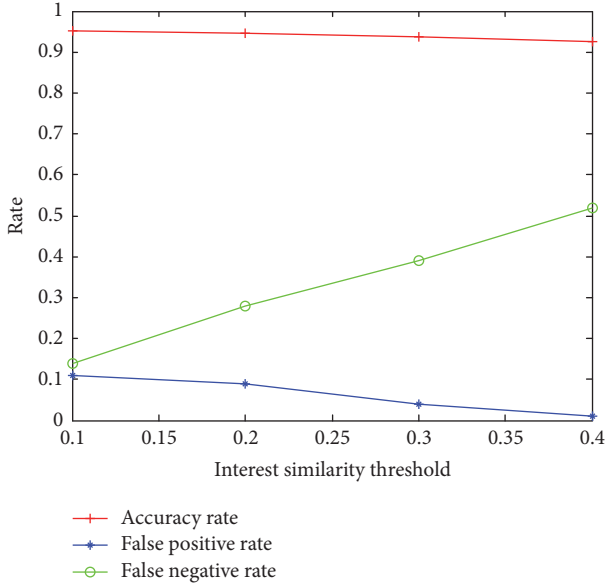


FIGURE 4: The false positive rate and false negative rate.

which helps to reduce the effects of poison attacks to some extent.

The false positive rate and false negative rate of CPSFS are shown in Figure 4. False negative represents spam that is classified as a legitimate email incorrectly. False positive represents a legitimate email that is classified as spam incorrectly. The interest similarity threshold is set to 0.1, 0.2, 0.3, and 0.4, respectively. The trust threshold is set to 0.5. The false positive rate decreases and the false negative rate increases as the interest similarity is increasing.

6.3. Different Trust Threshold and Interest Similarity Threshold.

For formula (2), the value of parameter b influences the trust calculation and the accuracy rate. In our evaluations, the value of b is set to 0.1, 0.2, and 0.3, respectively, and the accuracy rate results are shown in Figure 5. We can see that when the trust threshold is set to 0.5, all the three curves reach their accuracy peaks. In this figure, the accuracy rate changes dramatically as the trust threshold changes in all the three curves, which indicates that the trust threshold affects the accuracy significantly. When b is set to 0.1, the correlation between accuracy and trust is the best as shown in Figure 5. Therefore the value of b is set to 0.1 when calculating trust values.

The interest similarity threshold is a key factor which influences the performance of spam filtering. The results are shown in Figure 6, where the trust thresholds are set to 0.4, 0.5, 0.6, and 0.7, respectively. It shows that the accuracy rate changes when the interest similarity threshold increases from 0.1 to 0.4. The accuracy rate is always higher when the interest similarity threshold is set to 0.1 compared with other interest similarity thresholds. The accuracy decreases when the interest similarity threshold increases because users will not push their spam reports if the interest similarities are lower than the corresponding thresholds.

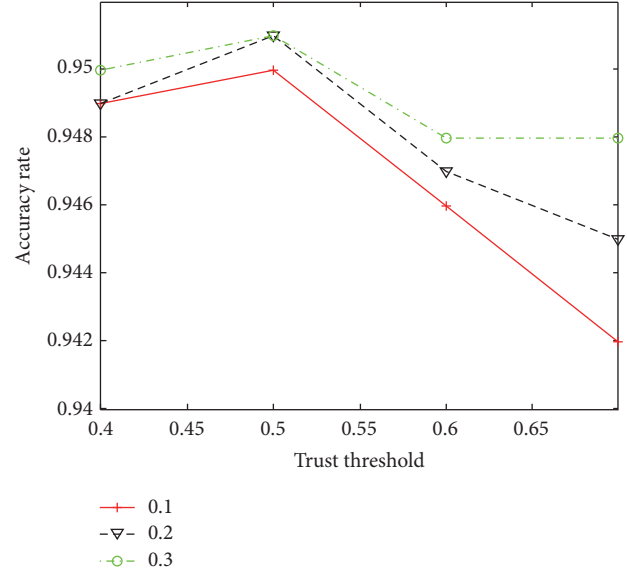


FIGURE 5: Accuracy rate under different trust threshold.

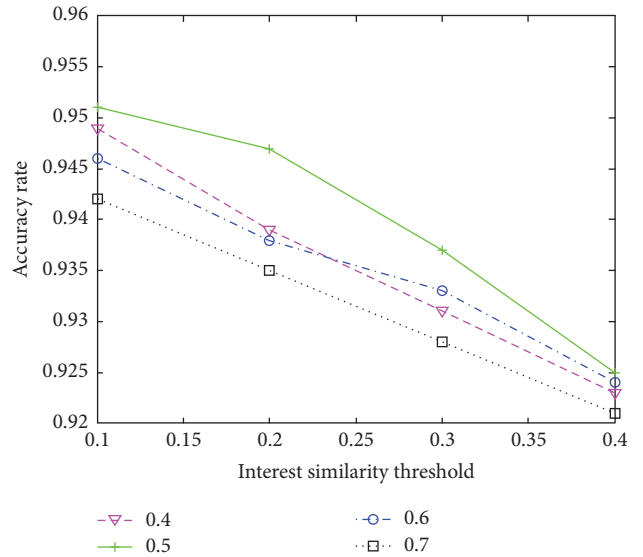


FIGURE 6: The accuracy rate under different interest similarity threshold.

7. Conclusion

To handle spam effectively, in this paper, we propose a credible and personalized spam filtering scheme (CPSFS) based on social trust and interest similarity, where users report their received spam emails to their contacts in social networks. We introduced local lists and social sensing mechanism for spam reports. The trust value and similarity are calculated to determine whether users should push spam reports to their friends. The social trust and similarity increase the credibility of the CPSFS filtering scheme. Our experiments showed that the accuracy of our CPSFS is better than the conventional Bayesian filter and some other approaches.

There is work on Copy Adjustable Incentive Scheme (CAIS) that adopts virtual credit concept to stimulate selfish nodes to cooperate in data forwarding [32]. We will consider incorporating this idea to the situation where some users are reluctant to share interests. In the future, we will also improve the performance by improving the network connectivity and throughput [33]. In addition, we will apply our CPSFS scheme to other social networks such as mobile social networking and vehicular social networks [34].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

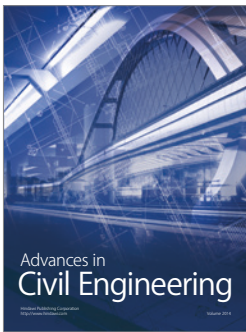
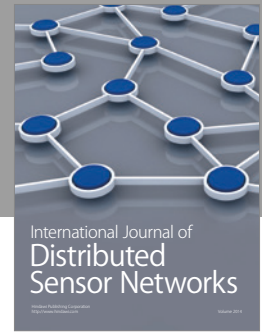
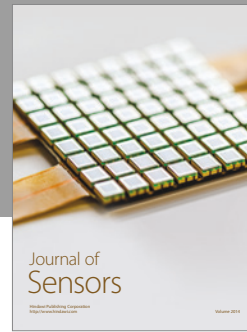
Acknowledgments

The work presented in this paper is supported by the Key Research Program of Shandong Province (no. 2017GGX10140), National Natural Science Foundation of China (no. 61309024, and no. 61772551), the Program on Innovative Methods of Work from Ministry of Science and Technology, China (no. 2015010300), Shandong Provincial Natural Science Foundation (no. ZR2015FM022), and the Fundamental Research Funds for the Central Universities.

References

- [1] Z. Miller, B. Dickinson, W. Deitrick, W. Hu, and A. H. Wang, "Twitter spammer detection using data stream clustering," *Information Sciences*, vol. 260, pp. 64–73, 2014.
- [2] J. Shen, R. H. Deng, Z. Cheng, L. Nie, and S. Yan, "On robust image spam filtering via comprehensive visual modeling," *Pattern Recognition*, vol. 48, no. 10, pp. 3227–3238, 2015.
- [3] B. Cui, Z. Liu, and L. Wang, "Key-Aggregate Searchable Encryption (KASE) for Group Data Sharing via Cloud Storage," *IEEE Transactions on Computers*, vol. 65, no. 8, pp. 2374–2385, 2016.
- [4] Z. Liu, X. Chen, J. Yang, C. Jia, and I. You, "New order preserving encryption model for outsourced databases in cloud environments," *Journal of Network and Computer Applications*, vol. 59, pp. 198–207, 2016.
- [5] *China Internet Association Anti-Spam center: Anti-spam survey report*, <http://www.acma.gov.au/theACMA/spam-statistics>.
- [6] Y.-M. Li, H.-W. Hsiao, and Y.-L. Lee, "Recommending social network applications via social filtering mechanisms," *Information Sciences*, vol. 239, pp. 18–30, 2013.
- [7] W. Mason, J. W. Vaughan, and H. Wallach, "Computational social science and social computing," *Machine Learning*, vol. 95, no. 3, pp. 257–260, 2014.
- [8] Y. Ren and D. Ji, "Neural networks for deceptive opinion spam detection: An empirical study," *Information Sciences*, vol. 385–386, pp. 213–224, 2017.
- [9] H. Shen, Z. Li, J. Liu, and J. E. Grant, "Knowledge sharing in the online social network of Yahoo! Answers and its implications," *IEEE Transactions on Computers*, vol. 64, no. 6, pp. 1715–1728, 2015.
- [10] Y. Ren, J. Shen, J. Wang, J. Han, and S. Lee, "Mutual verifiable provable data auditing in public cloud storage," *Journal of Internet Technology*, vol. 16, no. 2, pp. 317–323, 2015.
- [11] D. Quinn, L. Chen, and M. Mulvenna, "Social network analysis: A survey," *International Journal of Ambient Computing and Intelligence*, vol. 4, no. 3, pp. 46–58, 2012.
- [12] Y. A. Kim and M. A. Ahmad, "Trust, distrust and lack of confidence of users in online social media-sharing communities," *Knowledge-Based Systems*, vol. 37, pp. 438–450, 2013.
- [13] S. J. Yu, "The dynamic competitive recommendation algorithm in social network services," *Information Sciences*, vol. 187, no. 1, pp. 1–14, 2012.
- [14] F. Liu, X. Li, Y. Ding et al., "A social network-based trust-aware propagation model for P2P systems," *Knowledge-Based Systems*, vol. 41, pp. 8–15, 2013.
- [15] X. Liu, L. Shi, Y. Wang, Z. Xin, and W. Fu, "A dynamic trust conference algorithm for social network," in *Proceedings of the 2013 8th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 3PGCIC 2013*, pp. 340–346, France, October 2013.
- [16] X. Liu, Y. Wang, D. Zhao, W. Zhang, and L. Shi, "Patching by automatically tending to hub nodes based on social trust," *Computer Standards & Interfaces*, vol. 44, pp. 94–101, 2016.
- [17] J. Jaeyeon and S. Emil, "An Empirical Study of Spam Traffic and the Use of DNS Black Lists," in *Proceedings of the 4th ACM SIGCOMM Conference on Internet measurement*, pp. 370–375, October 2004.
- [18] C. O'Brien and C. Vogel, "Spam filters: Bayes vs. Chi-squared; letters vs," in *Proceedings of the in Proceedings of the 1st International Symposium on Information and Communication Technologies*, pp. 291–296, September 2003.
- [19] P. Haider, U. Brefeld, and T. Scheffer, "Supervised clustering of streaming data for email batch detection," in *Proceedings of the 24th International Conference on Machine Learning, ICML 2007*, pp. 345–352, USA, June 2007.
- [20] B. Scholkopf and J. B. Platt, "Dirichlet-Enhanced spam filtering based on biased samples," in *Proceedings of the International Conference on Neural Information Processing Systems*, pp. 161–168, MIT Press, January 2006.
- [21] D. Zisiadis, S. Kopsidas, A. Varalis, and L. Tassioulas, "Mailbook: A social network against spamming," in *Proceedings of the 2011 International Conference for Internet Technology and Secured Transactions, ICITST 2011*, pp. 245–249, are, December 2011.
- [22] P. O. Boykin and V. P. Roychowdhury, "Leveraging social networks to fight spam," *The Computer Journal*, vol. 38, no. 4, pp. 61–68, 2005.
- [23] M. Sirivianos, K. Kim, and X. Yang, "SocialFilter: Introducing social trust to collaborative spam mitigation," in *Proceedings of the IEEE INFOCOM 2011*, pp. 2300–2308, China, April 2011.
- [24] H. Shen and Z. Li, "Leveraging social networks for effective spam filtering," *IEEE Transactions on Computers*, vol. 63, no. 11, pp. 2743–2759, 2014.
- [25] C. Wilson, B. Boe, A. Sala, K. P. N. Puttaswamy, and B. Y. Zhao, "User interactions in social networks and their implications," in *Proceedings of the 4th ACM European Conference on Computer Systems, EuroSys'09*, pp. 205–218, Germany, April 2009.
- [26] M. Kobayakawa, S. Kinjo, M. Hoshi, T. Ohmori, and A. Yamamoto, "Fast computation of similarity based on jaccard coefficient for composition-based image retrieval," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 5879, pp. 949–955, 2009.
- [27] Z. Li and H. Shen, "SOAP: A Social network Aided Personalized and effective spam filter to clean your e-mail box," in *Proceedings of the IEEE INFOCOM 2011*, pp. 1835–1843, China, April 2011.

- [28] Z. Zhong and K. Li, "Speed up statistical spam filter by approximation," *IEEE Transactions on Computers*, vol. 60, no. 1, pp. 120–134, 2010.
- [29] S. Hameed, X. Fu, P. Hui, and N. Sastry, "LENS: Leveraging social networking and trust to prevent spam transmission," in *Proceedings of the 19th IEEE International Conference on Network Protocols, ICNP 2011*, pp. 13–18, Canada, October 2011.
- [30] Datatang., <http://www.datatang.com/data/796>.
- [31] "Spam Assassin," <http://spamassassin.apache.org/downloads.cgi?update=201504291720/>.
- [32] Z. Ning, L. Liu, F. Xia, B. Jedari, I. Lee, and W. Zhang, "CAIS: a copy adjustable incentive scheme in community-based socially aware networking," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, pp. 3406–3419, 2017.
- [33] Z. Ning, F. Xia, X. Hu, Z. Chen, and M. S. Obaidat, "Social-oriented adaptive transmission in opportunistic internet of smartphones," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 810–820, 2017.
- [34] Z. Ning, F. Xia, N. Ullah, X. Kong, and X. Hu, "Vehicular Social Networks: Enabling Smart Mobility," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 16–55, 2017.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

