Defending against Spam in Tagging Systems via Reputations

Eric Chang
Yale University

***Abstract*—Global Internet is witnessing a rapidly growing popularity of tagging services on the social networks, which enable people to share and tag different categories of resources. However, the current tagging systems face a serious problem — tag spam. In this paper, we propose SpamLimit — a novel social-enhanced reputation mechanism against spam in tagging systems. First, we propose a basic reputation mechanism that provides the personalized reputation estimates to each user in system. Our approach can impose severe and quick punishment to spammers but also provide an incentive to promote normal users sharing the correct tags. Because users can rank the tag search results with the reputation estimates of owners of resources, the results provided by spammers can be degraded to the end of search results. Then, we utilize friend relationships, the social nature of tagging systems, to enhance the basic reputation mechanism. Because the friends are all real-world acquaintances, these reliable companions can provide many referential experiences to users. This will help to improve both performance and convergence of SpamLimit. Finally, our experiment results illustrate that SpamLimit can effectively defend against tag spam and work better than the existing tag search models in tagging systems.**

I. Introduction

With the rapidly growing popularity of tagging services on the global Internet [34], [35], [37], people can share, tag and search different categories of resources in the tagging systems, for instance, photos in Flickr [3], URLs in Del.icio.us [2], videos in YouTube [6], and research papers in CiteUlike [1]. For a typical tagging system, each specific *resource* is annotated with some *tags*, and the relation $\langle resource, tag \rangle$ that annotates a resource with tag is called a *posting* which maintains the association between resource and tag. Many studies indicated that current tagging systems are vulnerable to *tag spam*: misleading tags that are generated to make it more likely that some resources are seen by users, or generated simply to confuse users [11], [16]. For instance, some *spammers* may repeatedly annotate some videos or photos with the erroneous tags, so that users searching for those tags will see an unexpected movies or photos. However, most of the previous works mainly focused on exploiting the potential search capability of tagging systems [7], [32], [12], and there is not any suitable solution on defending against tag spam yet.

This paper proposes SpamLimit — a novel social-enhanced reputation mechanism against tag spam for tagging systems. SpamLimit enables each user to assign others in the system with a *reputation estimate* so that the search results can be ranked by these reputations from the owners of resources. Thus, the results provided by spammers can be degraded to the end of search result pages. SpamLimit encompasses two key techniques to obtain the quality tag search results. First, it proposes for user two ways to compute the reputations of different participants in system: *unfamiliar users* and *interacted users*. For the former, SpamLimit can compute the reputation estimates among users with the statistical correlation of postings, and for the latter, our approach can reward or punish the result providers according to the evaluations on search results. SpamLimit can not only impose severe and quick punishment to spammers but also provide an incentive to promote normal users sharing the correct tags. Second, we utilize friend relationships, the social nature of tagging systems, to enhance the reputation calculations. In tagging systems, the friends are either acquaintances in reality or those online friends recognized in other social networks, thus these reliable companions can provide many referential experiences. Social-enhanced SpamLimit has better performance and convergence than the basic one on defending against tag spam.

We conducted simulation studies with some different configurations and compared SpamLimit with the existing tag search models, e.g., *Boolean* [5], *Occurrence* [4], and *Coincidence* [16]. The evaluation results show that SpamLimit can defend against tag spam from various attackers more effectively than the existing search models.

The rest of this paper is organized as follows. We give an overview of related work in Section II. Section III will describe the details of SpamLimit. The simulation methodology and evaluation results will be discussed in Section IV. Finally, we will conclude our work and present future work in Section V.

II. Related Work

A. Related Work on Tag Search Models

There are two prevalent tag search models in current tagging systems: Boolean model (e.g., Slideshare [5]) and Occurrence model (e.g., Rawsugar [4]). For Boolean model, the system randomly ranks the results associated with the search tag. For Occurrence model, the system ranks each resource based on the number of being annotated with the search tag and returns the top ranked resources.

B. Related Work on Defending against Tag Spam

Currently, some mechanisms have been proposed to address the problem of tag spam in tagging systems [33], [40], [20], [28], [29], [36], [30], [41], [38]. In general, these mechanisms can be grouped into three categories: *detection-based* mechanisms, *interface-based* mechanisms and *demotion-based* mechanisms. In detection-based mechanisms, the study in [17] investigated the usefulness of different machine learning algorithms and features, and transferred the approach based on determining relevant features to tagging systems to identify those spammers. Based on the notion that similar users and postings tend to use the same language, the study in [8] introduced *language model* to address the problem of spam in tagging systems. In interface-based mechanisms, CAPTCHAs [27] can be used to prevent automated account creation or automated tag spam posting. In demotion-based mechanisms, the study in [31] took into account spam by proposing a credibility score for each user based on the quality of the tags contributed by the user. The studies in [16], [11] proposed Coincidence model against spam in tagging systems and compared different rank methods for tag-based search models. To the best of our known, there is not any existing study that utilizes reputation mechanism or social-based scheme to defend against tag spam.

C. Related Work on Social Tagging Services

Except the above studies on defending against the tag spam, the increasing popularity of tagging system has motivated a number of studies [10], [18], [25], [31] that mainly focus on understanding tag usage and evolution. Recently, many companies were also trying to take advantage of the social tagging phenomenon inside the enterprise [9], [14], [21]. Moreover, the social knowledge in tagging system can be utilized to give participants some *tag recommendation* based on the interests and characterizing from the other users [22], [31], [24], [26], [39], [13].

III. Design Rationale of SpamLimit

In a tagging system, when user $Alice$ wants to acquire a specific resource R, he can issue a tag search t to the system, and then the system presents the result pages including the resources annotated with t to $Alice$. SpamLimit enables user to assign others in the system with a *reputation estimate* so that the search results can be ranked by these reputation estimates from the owners of resources. Here, SpamLimit has two ways to calculate the personalized estimates, one is for *unfamiliar users* and other is for *interacted users*. After browsing any resource in results, $Alice$ can evaluate the one by himself, and send feedback -1 or $+1$ to system, where -1 represents *unsuited result* and $+1$ means *correct result*.

Based on the above description, we define SpamLimit as two functional parts: *Assigning Reputation Estimates* and *Ranking Search Results*. The details of them will be described in the following Section III-A and Section III-B.

A. Assigning Reputation Estimates

In this section, we describe how SpamLimit computes the reputations for each participant in the system. We first present basic SpamLimit in Section III-A1, and then discuss social-enhanced SpamLimit in Section III-A2.

1) Basic SpamLimit: In basic SpamLimit, each user can assign a personal reputation estimate to every other participant in the system respectively. Generally, there are two different categories of users existing in the system, and SpamLimit provides two methods to compute the reputation estimates for them: unfamiliar users and interacted users. For any new comer, he will compute the initial reputation estimates to the others with the method of unfamiliar users, because that the users in system are all strange for him at this time.

Unfamiliar Users. For the users we have never interacted with, we do not have any referential experiences to assign the estimates. In this case, SpamLimit utilizes the statistical correlation of postings among users to calculate the reputation estimates with each other. We define that a pair of posting is *coincident* if and only if both the resources and the tags are the same respectively in two postings. Now, user a can calculate the reputation estimate $RE_{a,b}$ of the unfamiliar user b by examining the proportion of coincident postings between a and b.

$$RE_{a,b} = \frac{\sum\limits_{r_i \in CR} \sum\limits_{t_j \in CT} |post(r_i, t_j)|}{\sum\limits_{r_i \in CR} \sum\limits_{t_j \in T} |post(r_i, t_j)|} \tag{1}$$

where

- CR: the set of resources owned by user a and user b in common.
- CT: the set of coincident tags annotated by user a and user b to the same resources.
- T: the set of tags annotated by user b to the resources.
- $post(r_i, t_j)$: the set of postings annotated r_i with t_j.

In this case, user a can only compute accurate and strong reputation estimates for user b if he has himself cast a sufficient number of both inconsistent and coincident postings. This restriction provides a strong incentive mechanism for users to participate in annotating because users who do not annotate correctly will find the quality of the estimates they compute noticeably degraded. A user can still benefit from SpamLimit by tagging honestly but inactively, suppressing the sharing and dissemination of erroneous tags to the system.

Interacted Users. For the users we have interacted with, the calculations of reputation estimates are based on our previous experiences from them. In SpamLimit, after evaluating the quality of result from user b, system can update a's reputation estimate with b, $RE_{a,b}$, as follows:

$$RE_{a,b} = \begin{cases} max(-1, RE_{a,b} - pn^2) & \text{if result is unsuilt} \\ min(1, RE_{a,b} + r) & \text{otherwise} \end{cases} \tag{2}$$

where

- n: the number of consecutive discoveries of erroneous posting results from user b (including the last one).

- p: the penalty factor given to user b for each erroneous posting result evaluation.
- r: the recompense given to user b for each correct posting result evaluation.

Note that, in this case, the reputation estimate from a to b decreases faster than it increases. Aiming at severely penalizing spammers and those users who only occasionally share the correct postings, SpamLimit weights p by the square of the number of unsuited results discovery. Moreover, SpamLimit uses different penalty and recompense factors, and we propose to set $p > r$.

SpamLimit defines that parameter $RE_{a,min}$ is the minimum level of trust for any user by a. User a will not respond to the users who is currently considered untrustworthy. Due to refusing to respond a tag query, SpamLimit provides incentive for enhancing the honest annotating and sharing of participants in system.

2) Social-enhanced SpamLimit: In specific applications, SpamLimit may faces some problems. For instance, due to the lack of experiences or overlapping interest sets, a new comer may be unable to obtain the ideal search results with the basic SpamLimit. Another example, in a system which has the large scale number of users, basic SpamLimit can not provide the good convergence on obtaining quality search results. To address above practical issues, we hope to find the solution through exploring the characteristic of tagging systems. Considering about the social nature of tagging systems, we utilize the friend relationships of tagging systems to enhance the basic SpamLimit.

In social-enhanced SpamLimit, each user has many friends and stores their information in his own friend list. Any new comer can be invited by an existing user, and thus added into the system; meanwhile the latter will become the friend of the new comer automatically. Besides the method of invitation, we can establish friend relationships with the users who are our acquaintances in reality or some online friends recognized in some social networks, e.g., Facebook. SpamLimit utilizes the friend relationships based on the notion that friends are more reliable than those unknown users in network, and they can provide more authentic referential experiences which can be used as an important component for reputation calculation.

Here, user a can calculate reputation with respect to user b, $R_{a,b}$, as follows:

$$R_{a,b} = \beta RE_{a,b} + (1 - \beta) FE_{a,b} \tag{3}$$

where

- $RE_{a,b}$: user a's reputation estimate to user b computed by the basic SpamLimit.
- $FE_{a,b}$: the experiences from user a's friends with respect to user b.
- β: ($0 \leq \beta \leq 1$) controls the weights given to user a's reputation estimate and the friend experiences with respect to user b.

The experiences from friends, $FE_{a,b}$, are the average computing of the statistical correlation (using Formula (1)) of postings between each user a's friend and user b, referring to Formula (4).

$$FE_{a,b} = \frac{\sum_{f \in F} RE_{f,b}}{|F|} \tag{4}$$

where

- $RE_{f,b}$: the statistical correlation of postings between user a's friend f and user b, computed by Formula (1).
- F: the set of user a's friends.

Practical Issue. Practically, some friends may be deceivers or compromised. In order to avoid from the harm by these malicious friends, SpamLimit assigns reputation estimates to all friends of each user, and set this estimate to 1 by default. When user a evaluates -1 to a search result, besides decreasing the reputations of result providers, SpamLimit also examines whether user a's friends also share this erroneous result. If some friends do that indeed, their reputations will also be decreased according to the Formula (2). Note that, when we compute the experiences from friends, the calculation will not be affected by the reputation estimates of friends. After the reputation of user a's friend drops to below $RE_{a,min}$, a will disregard the experiences from this friend. Thus, this malicious friend can not continue providing erroneous results.

B. Ranking Search Results

For ranking search results, SpamLimit adopts the common method, using average computing to obtain the rank of every resource. User a can calculate rank with respect to resource r, $rank_{a,r}$, as follows:

$$rank_{a,r} = \frac{\sum_{u \in U} R_{a,u}}{|U|} \tag{5}$$

where

- $R_{a,u}$: the reputation of user a with respect to user u, computed by Formula (3).
- u: the user who has the resource r.
- U: the set of owners of resource r.

Now, when we issue the tag queries to the system, those unmatched results provided by spammers can be degraded to the end of search results.

IV. Evaluation

In this section, we first describe the simulation setup, and then we describe the behaviors of various attackers in practical tagging systems. Finally, we evaluate the performance of SpamLimit by comparing with the prevalent search models, Boolean [5], Occurrence [4] and Coincidence [16].

A. Simulation Setup

To evaluate the performance of SpamLimit, we developed a prototype of tagging system. Our prototype is an event-driven simulator which is composed of M users including good users and attackers. We simulate a large number of user transactions, where each user can search and get resources with SpamLimit. The transaction is a two step process for user: the tag query

TABLE I: Default Parameters in Experiments

Parameter	Value
Number of Resources R	50,000
Size of Vocabulary V	10,000
Number of Resources in Result $\mathcal{K}$	20

Parameter	Good Users	Attackers
Number of Users	800	200
Owned Resources (At the Startup)	30	50
Search Rate (resources / day)	0-10 / day	0

and the returned binary evaluation rating of -1 (erroneous) or $+1$ (correct). There are R resources in the system and V size of vocabulary for users to tag their resources. At the startup of simulation, the selections of resources and postings follow the distributions measured in [23] and [19] respectively.

We utilize Kleinberg model [15], a widely accepted social network model, to generate the friend relationships among users, and assign 12 friends to each user averagely according to the measurement results in [23]. At the startup of simulation, each good user annotates all his resources with the correct tags. Then, all the good participants in system search the tags they are interest in and choose some of them to browse or save according to the order of search result. Even if obtain the unmatched resources, they will annotate the ones with the correct tags. Thus, the good users have the correct postings only. Besides, a malicious user shares those misleading postings and participates in the system to spread them — this attempt to undermine the performance of system.

B. Threat Models

Here, we describe some models of malicious users in our simulation, and they are all the representative attackers in the practical tagging systems.

Random Attack Model. For the random attackers, they will select some of their resources and annotate them with some erroneous tags randomly in order to mislead those normal participants in the system. Normally, the random attack acts independently, that is, these bad users are "lousy taggers".

Targeted Attack Model. In some cases, the malicious collusive users may launch a particular form of targeted attack that colluding users attack a particular resource with the number of misleading tags or annotate their resources which they want to disseminate with the popular tags. This category of attack is very serious, and existing works have no good solution on this attack.

Disguised Attack Model. Some malicious users may launch a trickish attack which can avoid from being detected by the existing anti-spam mechanisms. They annotate their resources with both the correct and erroneous tags, and the existing anti-spam mechanisms will be a victim when encountering this attack. The attackers can be individual or collusive in the system.

C. Evaluation Results

We ran experiments with various configurations and obtained qualitatively similar results. Table I summarizes all parameters considered and their default values in next results discussed. Moreover, we set the parameters of SpamLimit, $p = 0.2$, $r = 0.1$, $\beta = 0.5$, and $R_{min} = 0.3$.

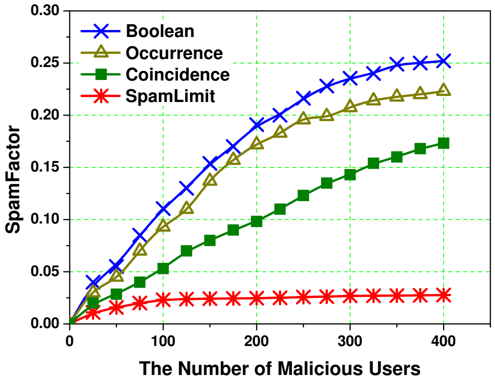

Fig. 1: Impact of the Number of Malicious Users

SpamFactor. Before discussing the experiment results, we first explain the metric for evaluating our experiments. Since our purpose is to evaluate the impact of tag spam on the search result, we make use of the metric *SpamFactor* [11], [16] accepted widely for quantifying the "spam impact" on search result. However, what should the factor be so as to let us obtain the search result as we desire? The study in [16] has argued that SpamFactor less than 0.1 is "tolerable" in the sense that the spam files will be few and towards the bottom of the result list. In our experiments, we are interested in the SpamFactors of the top $\mathcal{K}$ results.

Impact of the Number of Malicious Users. While keeping the number of total users as $1,000$ constantly, we vary the number of malicious users from 0 to 400. In this experiment, the malicious users are all random attackers. The result shown in Fig. 1 clearly indicates that both Boolean and Occurrence suffer from the attack of tag spam when the percentage of malicious users grow higher than 10%. The reason that the SpamFactors of Boolean and Occurrence increase higher than 0.1 so quickly is that their designs do not take spam problem into account. We observe that Coincidence model works better than above two models; however, after increasing the number of random attackers upon 200, the SpamFactor of Coincidence model is also higher than 0.1. This is because that, as the number of malicious users proliferate, the probability of coincident postings from malicious users also become higher. For SpamLimit, even if the numbers of malicious users become 400 in system, we can still control the SpamFactor lower than 0.05.

Impact of the Number of Users. In this experiment, we increase the number of total users in tagging system, and keep the percentage of random attackers in system maintaining 10%. Fig. 2 shows that, as the increasing with the number of users in system, only Boolean model has been keeping the SpamFactor higher than 0.1. This is because the random attackers can not launch the attacks to some particular tags

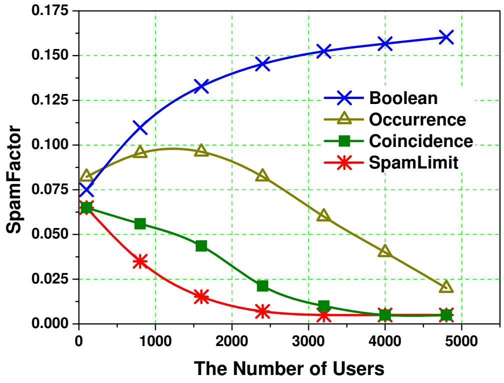

Fig. 2: Impact of the Number of Users

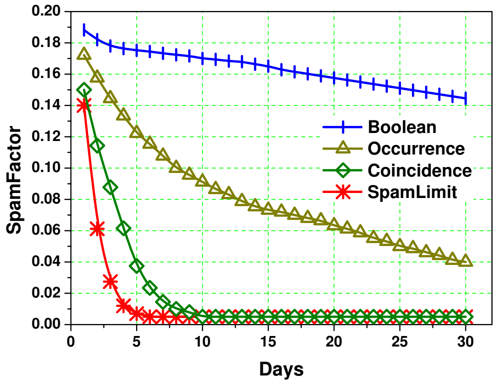

Fig. 3: Impact of the Random Attacks

collusively. Note that, the initial degradation of Occurrence model is due to the lack of the enough users in the system to generate the referential postings. Moreover, we can also observe that the SpamFactor of SpamLimit can quickly drop to 0.025 with the best convergence.

Impact of the Random Attacks. According to the parameters in Table I, we utilize simulator to compare both the performance and convergence of all the models in the environment of 20% random attackers. As shown in Fig. 3, the SpamFactor of Boolean model always maintains higher than 0.1 during our experiment. For other models, SpamLimit presents the best convergence because its SpamFactor can decrease below 0.1 in only 2 days. Coincidence model can also work well when there are 20% random attackers in the system, and its SpamFactor drops to 0.1 in 3 days. Due to the comparatively large percentage of attackers, the SpamFactor of Occurrence model can decrease below 0.1 in 8 days. That is said Occurrence model can work well under the random attacks but needs time to converge.

Impact of the Targeted Attacks. In this experiment, we study the impact of those collusive targeted attackers in tagging system. As shown in Fig. 4, we observe that the SpamFactor of Boolean model decreases gradually and becomes lower than 0.1 after 4 days. Comparing with Boolean, the SpamFactors of Occurrence model and Coincidence model are always upon 0.2 and 0.3 respectively during our experiment. Since the purpose of targeted attackers annotating their resources with those particular erroneous tags collusively is to drive up the number of the misleading postings, Occurrence and Coincidence can not work well when encountering this attack. However, Boolean model is to choose the resources randomly so it does not suffer from the targeted attacks significantly. We notice that the SpamFactor of SpamLimit can drop to 0.1 in 6 days and the performance of SpamLimit exceeds Boolean model after only 8 days.

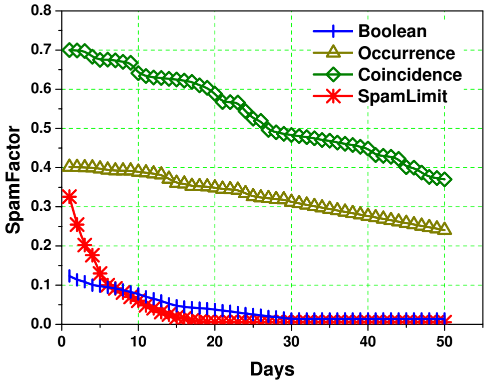

Fig. 4: Impact of the Targeted Attacks

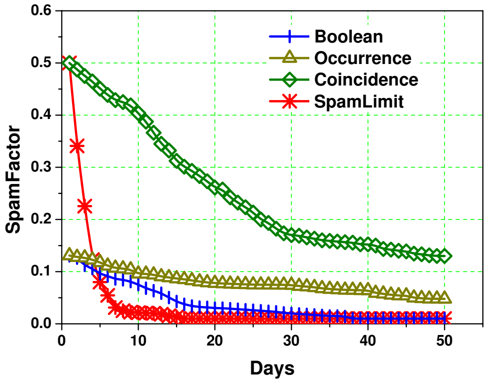

Fig. 5: Impact of the Disguised Attacks

Impact of the Disguised Attacks. In this experiment, we compare four models under the disguised attacks. As shown in Fig. 5, we observe that Boolean model and Occurrence model can work well when encountering the disguised attacks. We also notice that both SpamLimit and Coincidence can return the results with the high SpamFactors at the startup of this experiment. This is due to the lack of adequate interacted experiences; two models treat those disguised participants as the good ones. However, through utilizing the severe penalty mechanism, SpamLimit can converge quickly and decrease

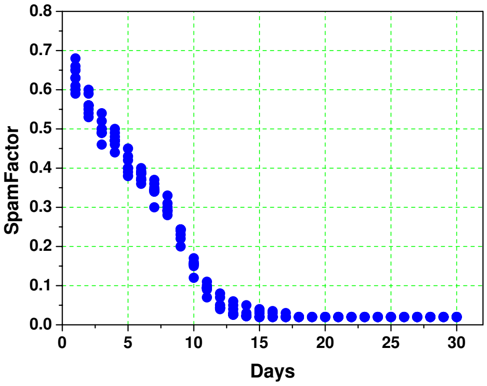

Fig. 6: Impact of the Non-experience Comers

SpamFactor below 0.1 in only 5 days. For Coincidence model, its SpamFactor always maintains upon 0.2. It is clear that the disguised attack can impact both SpamLimit and Coincidence model; however, SpamLimit can perform much more quickly based on its mechanisms than Coincidence.

Impact of the Non-experience Comers. In order to illustrate the capability for dealing with the non-experience comers of SpamLimit, we add 100 new comers into current network. In this experiment, there are 20% targeted attackers in the system. At startup, each of these 100 new comers has five friends, and all the new comers do not have any resource. As shown in Fig. 6, depending on the experiences provided by friends, the SpamFactors of these non-experience comers can sharply drop to 0.1 in about 12 days. This proves that SpamLimit can work well for the non-experience comers.

V. Conclusion and Future Work

Current tagging systems are highly vulnerable to tag spam. In this paper, we propose SpamLimit, a novel social-enhanced reputation mechanism against spam in tagging systems. In SpamLimit, each user can calculate the personal reputation estimates for others in the system, and utilize them to rank the search results so that the spam ones can be degraded to the end of the result pages. Moreover, SpamLimit utilizes social networks to enhance both the performance and convergence of defending against tag spam. As the experiment results shown, SpamLimit can work better than the existing search models on defending against tag spam.

There will be also much interesting work to be done in future. For instance, can we enhance SpamLimit through introducing the friend communities or interest-based groups? If the specific behaviors of users can be captured, shall we propose a new mechanism against tag spam based on the analysis of user behaviors?

References

[1] CiteUlike, http://www.citeulike.org/.

[2] Del.icio.us, http://del.icio.us/.

[3] Flickr, http://www.flickr.com/.

[4] Rawsugar, http://rawsugar.com/.

[5] Slideshare, http://slideshare.net/.

[6] Youtube, http://www.youtube.com/.

[7] Shenghua Bao, Gui Rong Xue, Xiaoyuan Wu, Yong Yu, Ben Fei, and Zhong Su. Optimizing web search using social annotations. In *WWW*, pages 501–510, 2007.

[8] T. Bogers and A. Bosch. Using language models for spam detection in social bookmarking systems. In *RSDC*, 2008.

[9] L. Damianos, J. Griffith, and D. Cuomo. Onomi: Social bookmarking on a corporate intranet. In *Collaborative Web Tagging Workshop in conjunction with WWW*, 2006.

[10] S. Golder and B. Huberman. Usage patterns of collaborative tagging systems. In *Journal of Information Science*, pages 198–208, 2006.

[11] Paul Heymann, Georgia Koutrika, and Hector Garcia-Molina. Fighting spam on social web sites: A survey of approaches and future challenges. *IEEE Internet Computing*, 11(6):36–45, 2007.

[12] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Information retrieval in folksonomies: Search and ranking. In *ESWC*, pages 411–426, 2006.

[13] Jianchun Jiang, Liping Ding, Ennan Zhai, and Ting Yu. Vrank: A context-aware approach to vulnerability scoring and ranking in SOA. In *Sixth International Conference on Software Security and Reliability, SERE 2012, Gaithersburg, Maryland, USA, 20-22 June 2012*, pages 61–70, 2012.

[14] A. John and D. Seligmann. Collaborated tagging and expertise in the enterprise. In *Collaborative Web Tagging Workshop in conjunction with WWW*, 2006.

[15] Jon M. Kleinberg. The small-world phenomenon: an algorithm perspective. In *STOC*, 2000.

[16] Georgia Koutrika, Frans Adjie Effendi, Zoltán Gyöngyi, Paul Heymann, and Hector Garcia-Molina. Combating spam in tagging systems. In *AIRWeb*, 2007.

[17] Beate Krause, Christoph Schmitz, Andreas Hotho, and Gerd Stumme. The anti-social tagger: detecting spam in social bookmarking systems. In *AIRWeb*, pages 61–68, 2008.

[18] R. Kumar, J. Novak, and A. Tomkins. Structure and evolution of online social networks. In *KDD*, 2006.

[19] Rui Li, Shenghua Bao, Yong Yu, Ben Fei, and Zhong Su. Towards effective browsing of large scale social annotations. In *WWW*, pages 943–952, 2007.

[20] Bo Liu, Ennan Zhai, Huiping Sun, Yelu Chen, and Zhong Chen. Filtering spam in social tagging system with dynamic behavior analysis. In *2009 International Conference on Advances in Social Network Analysis and Mining, ASONAM 2009, 20-22 July 2009, Athens, Greece*, pages 95–100, 2009.

[21] D. Millen, J. Feinberg, and B. Kerr. Social bookmarking in the enterprise. In *Social Computing*, 2005.

[22] Gilad Mishne. Autotag: a collaborative approach to automated tag assignment for weblog posts. In *WWW*, pages 953–954, 2006.

[23] Alan Mislove, Massimiliano Marcon, P. Krishna Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In *Internet Measurement Conference*, pages 29–42, 2007.

[24] T. Ohkura, Y. Kiyota, and H. Nakagawa. Browsing system for weblog articles based on automated folksonomy. In *Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics*, 2006.

[25] S. Sen, S. Lam, A. Rashid, D. Cosley, D. Frankowski, J. Osterhouse, F. Maxwell Harper, and J. Riedl. Tagging, communities, vocabulary, evolution. In *CSCW*, 2006.

[26] Cong Sun, Ennan Zhai, Zhong Chen, and Jianfeng Ma. A multi-compositional enforcement on information flow security. In *Information and Communications Security - 13th International Conference, ICICS 2011, Beijing, China, November 23-26, 2011. Proceedings*, pages 345–359, 2011.

[27] Luis von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford. Captcha: Using hard ai problems for security. In *EUROCRYPT*, pages 294–311, 2003.

[28] Yonggang Wang, Ennan Zhai, Cui Cao, Yongqiang Xie, Zhaojun Wang, Jian-bin Hu, and Zhong Chen. Dspam: Defending against spam in tagging systems via users' reliability. In *16th IEEE International Conference on Parallel and Distributed Systems, ICPADS 2010, Shanghai, China, December 8-10, 2010*, pages 139–146, 2010.

[29] Yonggang Wang, Ennan Zhai, Jian-bin Hu, and Zhong Chen. Claper: Recommend classical papers to beginners. In *Seventh International*

Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2010, 10-12 August 2010, Yantai, Shandong, China, pages 2777–2781, 2010.

[30] Yonggang Wang, Ennan Zhai, Eng Keong Lua, Jian-bin Hu, and Zhong Chen. isac: Intimacy based access control for social network sites. In *9th International Conference on Ubiquitous Intelligence and Computing and 9th International Conference on Autonomic and Trusted Computing, UIC/ATC 2012, Fukuoka, Japan, September 4-7, 2012*, pages 517–524, 2012.

[31] Z. Xu, Y. Fu, J. Mao, and D. Su. Towards the semantic web: Collaborative tag suggestions. In *Collaborative Web Tagging Workshop in conjunction with WWW*, 2006.

[32] Yusuke Yanbe, Adam Jatowt, Satoshi Nakamura, and Katsumi Tanaka. Towards improving web search by utilizing social bookmarks. In *ICWE*, pages 343–357, 2007.

[33] Ennan Zhai, Ruichuan Chen, Zhuhua Cai, Long Zhang, Eng Keong Lua, Huiping Sun, Sihan Qing, Liyong Tang, and Zhong Chen. Sorcery: Could we make P2P content sharing systems robust to deceivers? In *Proceedings P2P 2009, Ninth International Conference on Peer-to-Peer Computing, 9-11 September 2009, Seattle, Washington, USA*, pages 11–20, 2009.

[34] Ennan Zhai, Ruichuan Chen, David Isaac Wolinsky, and Bryan Ford. An untold story of redundant clouds: making your service deployment truly reliable. In *Proceedings of the 9th Workshop on Hot Topics in Dependable Systems, HotDep 2013, Farmington, Pennsylvania, USA, November 3, 2013*, pages 3:1–3:6, 2013.

[35] Ennan Zhai, Ruichuan Chen, David Isaac Wolinsky, and Bryan Ford. Heading off correlated failures through independence-as-a-service. In *11th USENIX Symposium on Operating Systems Design and Implementation, OSDI '14, Broomfield, CO, USA, October 6-8, 2014.*, pages 317–334, 2014.

[36] Ennan Zhai, Liping Ding, and Sihan Qing. Towards a reliable spam-proof tagging system. In *Fifth International Conference on Secure Software Integration and Reliability Improvement, SSIRI 2011, 27-29 June, 2011, Jeju Island, Korea*, pages 174–181, 2011.

[37] Ennan Zhai, Liang Gu, and Yumei Hai. A risk-evaluation assisted system for service selection. In *2015 IEEE International Conference on Web Services, ICWS 2015, New York, NY, USA, June 27 - July 2, 2015*, pages 671–678, 2015.

[38] Ennan Zhai, Zhenhua Li, Zhenyu Li, Fan Wu, and Guihai Chen. Resisting tag spam by leveraging implicit user behaviors. *PVLDB*, 10(3):241–252, 2016.

[39] Ennan Zhai, Qingni Shen, Yonggang Wang, Tao Yang, Liping Ding, and Sihan Qing. Secguard: Secure and practical integrity protection model for operating systems. In *Web Technologies and Applications - 13th Asia-Pacific Web Conference, APWeb 2011, Beijing, China, April 18-20, 2011. Proceedings*, pages 370–375, 2011.

[40] Ennan Zhai, Huiping Sun, Sihan Qing, and Zhong Chen. Spamclean: Towards spam-free tagging systems. In *Proceedings of the 12th IEEE International Conference on Computational Science and Engineering, CSE 2009, Vancouver, BC, Canada, August 29-31, 2009*, pages 429–435, 2009.

[41] Ennan Zhai, Huiping Sun, Sihan Qing, and Zhong Chen. Sorcery: Overcoming deceptive votes in P2P content sharing systems. *Peer-to-Peer Networking and Applications*, 4(2):178–191, 2011.