ISSN: 2321-7782 (Online)
Volume 3, Issue 12, December 2015
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Temporal Assosiations and Discovery of Ranking Fraud for Mobile Apps in Multivariate Time Series

Dr. K. Swathi[1]
Professor and Principal,
Department of Computer science &Engineering,
Cauvery College of Engineering & Technology, Trichy (TN)

T. Divya[2]
Scholar,
Department of Computer science &Engineering,
Cauvery College of Engineering & Technology, Trichy (TN)

Abstract: Cloud computing has recently emerged as a promising hosting platform that allows multiple cloud users called tenants to share a common physical computing infrastructure. With rapid adoption of the concepts of Software as a Service (SaaS) and Service Oriented Architecture (SOA), the Internet has evolved into an important service delivery infrastructure instead of merely providing host connectivity. In this project, we present IntTest, attestation scheme that can dynamically verify the integrity of data processing results in the cloud infrastructure and pinpoint malicious service providers when inconsistent results are detected. We validate service integrity by analyzing result consistency information with graph analysis. We proposed a new runtime service integrity attestation scheme that employs a novel attestation graph model to capture attestation results among different cloud nodes. We design attestation graph analysis algorithm to pinpoint malicious service providers and recognize colluding attack patterns. Our scheme can achieve runtime integrity attestation for cloud dataflow processing services using a small number of attestation data. Thus, our approach does not require trusted hardware or secure kernel co-existed with third-party service providers in the cloud. IntTest can achieve better scalability and higher detection accuracy than the state-of-the-art schemes. We extend our work in SAAS system to recommend links with improved trust results and with large number of service functions and services with consistency graph.

Keywords: Cloud computing, Secured Data Processing

I. INTRODUCTION

The number of mobile Apps has grown at a breathtaking rate over the past few years. For example, as of the end of April 2013, there are more than 1.6 million Apps at Apple's App store and Google Play. To stimulate the development of mobile Apps, many App stores launched daily App leader boards, which demonstrate the chart rankings of most popular Apps. Indeed, the App leader board is one of the most important ways for promoting mobile Apps. A higher rank on the leader board usually leads to a huge number of downloads and million dollars in revenue. Therefore, App developers tend to explore various ways such as advertising campaigns to promote their Apps in order to have their Apps ranked as high as possible in such App leaderboards. However, as a recent trend, instead of relying on traditional marketing solutions, shady App developers resort to some fraudulent means to deliberately boost their Apps and eventually manipulate the chart rankings on an App store. This is usually implemented by using so-called "*bot farms*" or "*human water armies*" to inflate the App downloads and ratings in a very short time. For example, an article from VentureBeat reported that, when an App was promoted with the help of ranking manipulation, it could be propelled from number 1,800 to the top 25 in Apple's top free leader board and more than 50,000-100,000 new users could be acquired within a couple of days. In fact, such ranking fraud raises great concerns to the mobile App industry. For example, Apple has warned of cracking down on App developers who commit ranking fraud in the Apple's App store. In the literature, while there are some related works, such as web ranking spam detection online review spam detection and mobile App recommendation the problem of detecting ranking fraud for mobile Apps is still under-explored. To fill this crucial void, in this paper, we propose to develop a ranking fraud detection system for mobile Apps. Along this line, we

identify several important challenges. First, ranking fraud does not always happen in the whole life cycle of an App, so we need to detect the time when fraud happens. Second, due to the huge number of mobile Apps, it is difficult to manually label ranking fraud for each App, so it is important to have a way to automatically detect ranking fraud without using any benchmark information. Finally, due to the dynamic nature of chart rankings, it is not easy to identify and confirm the evidences linked to ranking fraud. Indeed, our careful observation reveals that fraudulent Apps do not always be ranked high in the leaderboard, but only in some *leading events*, which form different *leading sessions*. Note that we will introduce both leading events and leading sessions in detail later. In other words, ranking fraud usually happens in these leading sessions. Therefore, detecting ranking fraud of mobile Apps is actually to detect ranking fraud within leading sessions of mobile Apps. Specifically, we first propose a simple yet effective algorithm

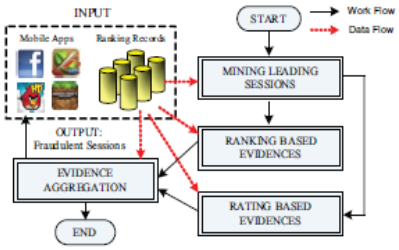


Figure 1: The framework of the ranking fraud detection system for mobile Apps.

To identify the leading sessions of each App based on its historical ranking records. Then, with the analysis of Apps' ranking behaviors, we find that the fraudulent Apps often have different ranking patterns in each leading session compared with normal Apps. Thus, we characterize some fraud evidences from Apps' historical ranking records, and develop three functions to extract such ranking based fraud evidences. Nonetheless, the ranking based evidences can be affected by some legitimate marketing campaigns, such as "limited-time discount". As a result, it is not sufficient to only use ranking based evidences. Therefore, we further propose two functions to discover rating based evidences, which reflect some anomaly patterns from Apps' historical rating records. In addition, we develop an unsupervised evidence aggregation method to integrate these two types of evidences for evaluating the credibility of leading sessions from mobile Apps. Figure 1 shows the framework of our ranking fraud detection system for mobile Apps. It is worth noting that all the evidences are extracted by modeling Apps' ranking and rating behaviors through statistical hypotheses tests. The proposed framework is scalable and can be extended with other domain-generated evidences for ranking fraud detection. Finally, we evaluate the proposed system with real-world App data collected from the Apple's App store for a long time period. Experimental results show the effectiveness of the proposed system, the scalability of the detection algorithm as well as some regularity of ranking fraud activities.

II. Related Work

N. Spirin and J. Hansome global information such as a large web graph and snapshots of a large Search engines became a de facto place to start information acquisition on the Web. Though due to web spam phenomenon, search results are not always as good as desired. Moreover, spam evolves that makes the problem of providing high quality search even more challenging. Over the last decade research on adversarial information retrieval has gained a lot of interest both from academia and industry. In this paper we present a systematic review of web spam detection techniques with the focus on algorithms and underlying principles. We categorize all existing algorithms into three categories based on the type of information they use: content-based methods, link-based methods, and methods based on non-traditional data such as user behaviour, clicks, HTTP sessions. In turn, we perform a subcategorization of link-based category into five groups based on ideas and principles used: labels propagation, link pruning and reweighting, labels refinement, graph regularization, and featurebased. We also define the concept of web spam numerically and provide a brief survey on various spam forms. Finally, we summarize the observations and underlying principles applied for web spam detection.

Demerits: web spam phenomenon, search results are not always as good as desired.

B. Zhou, J. Pei, and Z. Tang.Web spam, which refers to any deliberate actions bringing to selected web pages an unjustifiable favorable relevance or importance, is one of the major obstacles for high quality information retrieval on the web. Most of the existing web spam detection methods are supervised that require a large and representative training set of web pages. Moreover, they often assume collection of web pages. However, in many situations such assumptions may not hold.

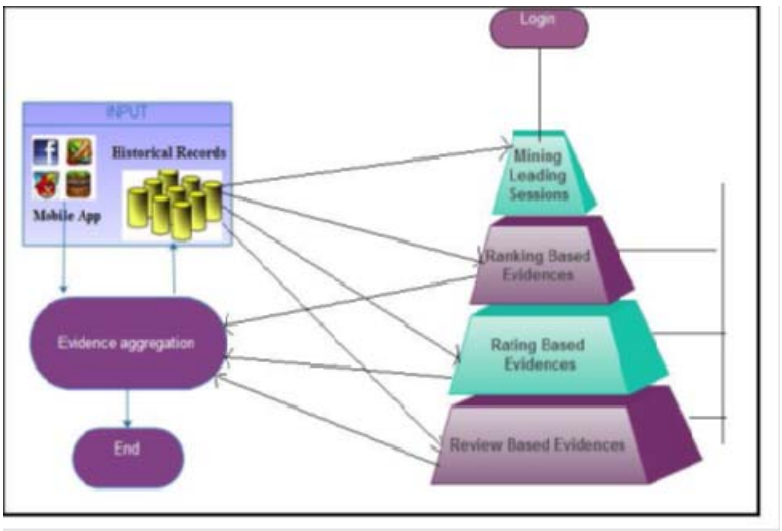
In this paper, we study the problem of unsupervised web spam detection. We introduce the notion of spamicity to measure how likely a page is spam. Spamicity is a more flexible and user-controllable measure than the traditional supervised classification methods. We propose efficient online link spam and term spam detection methods using spamicity. Our methods do not need training and are cost effective. A real data set is used to evaluate the effectiveness and the efficiency of our methods.

- **Demerits:** there is some subtlety about modelling web spam detection as a traditional classification problem.
- H. Zhu, E. Chen, K. Yu, H. Cao, H. Xiong, and J. Tian,we illustrate how to extract personal context-aware preferences from the context-rich device logs (i.e., context logs) for building novel personalized context-aware recommender systems. A critical challenge along this line is that the context log of each individual user may not contain sufficient data for mining his/her context-aware preferences. Therefore, we propose to first learn common context-aware preferences from the context logs of many users. Then, the preference of each user can be represented as a distribution of these common context-aware preferences. Specifically, we develop two approaches for mining common context-aware preferences based on two different assumptions, namely, context independent and context dependent assumptions, which can fit into different application scenarios. Finally, extensive experiments on a real-world data set show that both approaches are effective and outperform baselines with respect to mining personal context-aware preferences for mobile users.
- **Demerits:** individual user usually does not contain sufficient training information.

Modules

1. Leading events

Given a positioning limit K*∈[1, K] a main occasion e of App a contains a period range also, relating rankings of a, Note that positioning edge K * is applied which is normally littler than K here on the grounds that K may be huge (e.g., more than 1,000), and the positioning records past K * (e.g., 300) are not exceptionally helpful for recognizing the positioning controls. Moreover, it is finding that a few Apps have a few nearby driving even which are near one another and structure a main session.



A Framework of our ranking fraud detection system for Mobile App

2. Leading Sessions

Instinctively, mainly the leading sessions of mobile app signify the period of popularity, and so these leading sessions will comprise of ranking manipulation only. Hence, the issue of identifying ranking fraud is to identify deceptive sessions. Along with the main task is to extract the leading sessions of a mobile App from its historical ranking records.

Algorithm: Mining Leading Sessions: There are two main steps for mining leading sessions. First, we need to discover leading events from the Apps historical ranking records. Second, we need to merge adjacent leading events for constructing leading sessions.Specifically, Algorithm 1 demonstrates the pseudo code of mining leading sessions for a given App a.

Algorithm 1 Mining Leading Sessions

Input 1: a's historical ranking records R_a;
Input 2: the ranking threshold K^*;
Input 2: the merging threshold ϕ;
Output: the set of a's leading sessions S_a;
Initialization: $S_a = \emptyset$;

1: $E_s = \emptyset$; $e = \emptyset$; $s = \emptyset$; $t^e_{start} = 0$;
2: for each $i \in [1, |R_a|]$ do
3: if $r^a_i \leq K^*$ and $t^e_{start} == 0$ then
4: $t^e_{start} = t_i$;
5: else if $r^a_i > K^*$ and $t^e_{start} \neq 0$ then
6: //found one event;
7: $t^e_{end} = t_{i-1}$; $e =< t^e_{start}, t^e_{end} >$;
8: if $E_s == \emptyset$ then
9: $E_s \cup = e$; $t^s_{start} = t^e_{start}$; $t^s_{end} = t^e_{end}$;
10: else if $(t^e_{start} - t^s_{end}) < \phi$ then
11: $E_s \cup = e$; $t^s_{end} = t^e_{end}$;
12: else then
13: //found one session;
14: $s =< t^s_{start}, t^s_{end}, E_s >$;
15: $S_a \cup = s$; $s = \emptyset$ is a new session;
16: $E_s = \{e\}$; $t^s_{start} = t^e_{start}$; $t^s_{end} = t^e_{end}$;
17: $t^e_{start} = 0$; $e = \emptyset$ is a new leading event;
18: return S_a

III. Ranking Based Evidences

The ranking based evidences are useful for ranking fraud detection. However, sometimes, it is not sufficient to only use ranking based evidences. For example, some Apps created by the famous developers, such as Game loft, may have some leading events with large values of due to the developers' credibility and the "word-of mouth"advertising effect. Moreover, some of the legal marketing services, such as "limited-time discount", may also result in significant ranking based evidences. To solve this issue, we also study how to extract fraud evidences from Apps' historical rating records. According to the fact the definitions introduced in Section 2, a leading session is composed of several leading the events. Therefore, we should first analyze the basic characteristics of leading events for extracting fraud evidences. By analyzing the Apps' historical ranking records, we observe that Apps' ranking behaviours in a leading event always satisfy a specific ranking pattern, which consists of three the different ranking phases, namely, pising phase, maintaining phase and recession phase. Specifically, in each leading event, an App's ranking first increases to a peak position in the leader board (i.e., rising phase), then keeps those such peak position for a period (i.e., maintaining phase), and finally decreases till the end of the event (i.e., recession phase). Figure shows an example of different ranking phases of a leading event. T Indeed, such a ranking pattern shows an important understanding of leading event. In the following, we form ally define the three ranking phases of a leading event. Therefore, for both App developers and marketing Firms, the earlier the ranking expectation meets, the more Money can be earned. Moreover, the after reaching maintaining the expected ranking for a required period, the manipulation will be stopped and the ranking of t malicious App will decrease dramatically. As a result, the suspicious leading events may contain very short rising and recession phases.

Meanwhile, the cost of ranking manipulation with high ranking expectations is quite expensive due to unclear ranking principles of App stores and the fierce competition between App developers. Therefore, the leading event of fraudulent Apps often has very short maintaining phase with high ranking positions. Figure (a) shows an example of ranking records from one of the reported suspicious Apps [5]. We can see that his App has several impulsive leading events with high ranking positions. In contrast, the ranking behaviours of abnormal App's leading event may be completely different. For example, Figure 4 (b) shows an example of ranking records from a popular th App "Angry Birds: Space", which contains a leading event with a long time range (i.e. more than one year), especially for the recession phase. In fact, once a normal App is ranked high in the leader board, it often owns lots of honest fans and may attract More and more users the to download. Therefore, this App will be ranked high in hence the leader board for a long time. Based on the above discussion, hence we propose some ranking based signatures of leading sessions to construct fraud Evidences for ranking fraud detection

EVIDENCE 1.As shown in Figure 3, we use two shape parameters θ1 and θ2 to quantify the ranking patterns of the rising phase and the recession phase o App a's leading event e, which can be computed by

$$\mathbb{P}(\mathcal{P}(\lambda_s) \geq |E_s|) = 1 - e^{-\lambda_s} \sum_{i=0}^{|E_s|} \frac{(\lambda_s)^i}{i!}.$$

Where $K\_$ is the ranking threshold in Definition 1. Intuitively, a large $\theta 1$ may indicate that the App h been bumped to a high rank within a short time, and a large $\theta 2$ may indicate that the App has dropped from a high rank the bottom within a short time. Therefore, a leading session, which has more leading events with large $\theta 1$and $\theta 2$ values, has higher probability of having ranking fraud. He we define a fraud signature $\theta s$for a leading session as follows.

$$\overline{\theta}_s = \frac{1}{|E_s|} \sum_{e \in s} (\theta_1^e + \theta_2^e),$$

Where $|Es|$ is the number of leading events in session s. Intuitively, if a leading session s contains significantly Higher θscompared with other leading sessions of Apps In the leader board, it has high probability of having Ranking fraud. To capture this, hence we propose to apply Statistical hypothesis test for computing t heth significance Of $\theta s$for each leading session. We specifically, we define Two statistical hypotheses as follows and compute the P-value of each leading session.

$$\mathbb{P}(\mathcal{N}(\mu_{\overline{\theta}}, \sigma_{\overline{\theta}}) \geq \overline{\theta}_s) = 1 - \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{\overline{\theta}_s - \mu_{\overline{\theta}}}{\sigma_{\overline{\theta}}\sqrt{2}}\right)\right),$$

Where erf(x) is the Gaussian Error Function as follows,

Intuitively, a leading session with a smaller p-value P has more chance to r eject the HYPOTHESIS 0 and accept

HYPOTHESIS 1. This means it has more chance of committing ranking fraud. Thus, we define the evidence as

$$\Psi_1(s) = 1 - \mathbb{P}(\mathcal{N}(\mu_{\overline{\theta}}, \sigma_{\overline{\theta}}) \geq \overline{\theta}_s).$$

EVIDENCE 2

The number of leading events in a leading session, i.e.,$|Es|$, is also a strong signature ranking fraud. For a normal App, the recession phase indicates the fading of popularity. Therefore, after the end of a leading event, it is unlikely to appear another leading event in a short time unless the App updates its version or carries out some sales promotion. Therefore, if a leading session contains much more leading events compared with other leading sessions of Apps in the Leader board, it has high probability of having ranking fraud. To capture this , we define two statistical hypotheses t compute the significance of $|Es|$ for each leading session as follows.

» HYPOTHESIS 0*: The signature |Es| of leading session s is not useful for detecting ranking fraud.*

» HYPOTHESIS 1*: The signature |Es| of leading session s is significantly lager than expectation.*

Since |*Es*| always has discrete values, we propose to Leverage the Poisson approximation to calculate the p value With the above hypotheses. Specifically, we assume |*Es*| follows the Poisson distribution, |*Es*| ∼*P*(*λs*), where the parameter *λs*can be learnt by the MLE method from the observations of |*Es*| in all Apps' historical leading sessions. Then, we can calculate the p-value as follows,

$$\chi_s = \frac{1}{|E_s|} \sum_{e \in s} \frac{K^* - \overline{r}_m^e}{\Delta t_m^e},$$

IV. Rating Based Evidences

The ranking based evidences are useful for ranking fraud detection. However, sometimes, it is not sufficient to only use ranking based evidences. For the example, some Apps created by the famous developers, such as Game loft, may have some leading events with large values of $\theta 1$ due to the developers' credibility and the "word-of mouth" advertising effect. Moreover, some of the legal marketing services, such as "limited time discount" , may also result in significant ranking based evidences. To solve this issue, we also study how to extract fraud evidences from Apps' historical rating records. Specifically, after an App has been published, it can Be rated by any user who downloaded it. Indeed, user rating is one of the most important features of App advertisement. An App which has higher rating may attract more users to download and can also be ranked higher in the leader board. Thus, rating manipulation is also an important perspective Of ranking fraud. Intuitively, if an App has ranking fraud in a leading session *s*, the ratings during the time period of *s* may have anomaly patterns compared with its historical ratings, hen which can be used for constructing rating base evidences. For example, show the distributions of the daily average rating of a popular App "face book "Whats App" an a suspicious App discovered by our approach, respectively

EVIDENCE 1

For a normal App, the average rating in a specific leading session should be consistent with the aveage value of all historical ratings. In contrast, an App with rating manipulation might have surprisingly high ratings in the fraudulent leading sessions with respect to its historical ratings. Here, we define a fraud signature Δ*Rs*for each leading session as follows,

$$\Delta \mathcal{R}_s = \frac{\overline{\mathcal{R}}_s - \overline{\mathcal{R}}_a}{\overline{\mathcal{R}}_a}, \quad (s \in a)$$

where*Rs*is the average rating in leading session *s*, and *Ra* is the average historical rating of App *a*. Therefore, if a leading session has significantly higher value of Δ*R s* compared with other leading sessions of Apps in the leader board , it has high probability of having ranking fraud. To capture this we define statistical hypotheses to compute the significance of Δ*Rs* for each leading session as follows.

HYPOTHESIS 0*: The signature ΔRs of leading session s is not useful for detecting ranking fraud.*

HYPOTHESIS 1*: The signature ΔRs of leading session s is significantly higher than expectation.*

Here, we use the Gaussian approximation to calculate the p-value with the above hypotheses. Specifically, we assume Δ*Rs* follows the Gaussian distribution, Δ*Rs* ∼*N*(*μR, σR*), where *μR*and *σR*can be learnt by the MLE method from the observations of Δ*Rs* in a ll Apps' historical leading sessions. Then, we can compute the evidence by

$$\Psi_4(s) = 1 - \mathbb{P}(\mathcal{N}(\mu_{\mathcal{R}}, \sigma_{\mathcal{R}}) \geq \Delta \mathcal{R}_s).$$

V. REVIEW BASED EVIDENCES

Besides ratings, most of the App stores also allow users to write some textual comments as App reviews. Such reviews can reflect the personal perceptions and usage experiences of existing users for particular mobile Apps. Indeed, review manipulation is one of the most important perspectives of App ranking fraud. Specification before downloading or purchasing new mobile users often firstly read its historical reviews to ease their decision making, and a mobile App contains more positive reviews may attract more users to download.

Therefore, imposters often post fake reviews in the leading sessions of a specific App in order to inflate the App downloads, and thus propel the App's ranking position the leader board. Although some previous works overview spam detection have been reported in recent years [14], [19], [21], the problem of detecting the local anomaly of reviews in the leading sessions and capturing them as evidences for ranking fraud detection are still under-explored. To this end, here we propose two evidences based on Apps' review behaviours in leading sessions for detecting ranking fraud.

EVIDENCE

Indeed, most of the the review manipulations are implemented by bot farms due to the highcost of human resource. Therefore, review spamers often post multiple duplicate or near-duplicate reviews on the same App to inflate downloads [19], [21]. In contrast,the normal App always have diversified reviews sinceusers have different personal perceptions and usageexperiences. Based on the above observations, here wdefine a fraud signature *Sim*(*s*), which denotes the average mutual similarity between the reviews within leading thsession*s*. Specifically, this fraud signature canbe computed by following steps.

First, for each review *c* in leading session *s*, we remove all stop words (e.g., "of", "the") and normalize verbs and adjectives (e.g., "*plays* → *play*", "*better* → *good*"). Second we build a normalized words vector $-\!\!\rightarrow wc=dim[n]$ For each review *c* , where *n* indicates the number of all unique normalized words in all reviews of *s*. The specific, here we have $dim[i] = \Sigma freqi;cfreqi;c(1 \leq i \leq n)$,where *frequency* is the frequency of the *i*-th word in *c*.Finally, we can calculated the similarity between two reviews *ci* and *cj*by the Cosine similarity $Cos(-\!\!\rightarrow wci, -\!\!\rightarrow wcj)$.Thus, the fraud signature *Sim*(*s*) can be computed by

$$Sim(s) = \frac{2 \times \sum_{1 \leq i < j \leq N_s} Cos(\overrightarrow{w_{ci}}, \overrightarrow{w_{cj}})}{N_s \times (N_s - 1)},$$

Where *Ns* is the number of reviews during leading session *s*. intuitively, the higher value of *sin*(*s*) indicates more duplicate / near- duplicate reviews in *s*. Thus, if a leading session has significantly higher value of *Sin*(*s*) compared with other leading sessions of Apps in the leader board, it ha high probability of having ranking fraud. To capture this, we define statistical hypotheses to compute the significance of *Sin*(*s*) for each leading session as follows.

HYPOTHESIS 0*: The signature Sim*(*s*) *of leading session s is not useful for detecting ranking fraud.*

HYPOTHESIS 1*: The signature Sim*(*s*) *of leading session s is significantly higher than expectation.*

Here, we use the Gaussian approximation to compute the p-value with the above hypotheses. Specifically, we assume *Sin*(*s*) follows the Gaussian distribution, *Sin*(*s*) ∼*N*(*μSin, σSin*), where *μSin*and *σSin*can be learnt by the MLE method from the observations of *Sin*(*s*) in all Apps' historical leading sessions. Then, we can compute the evidence by

References

1. http://en.wikipedia.org/wiki/cohen's kappa.
2. https://developer.apple.com/news/index.php?id=0- 2062012a.
3. http://venturebeat.com/2012/07/03/apples-crackdown-on- app-ranking-manipulation/.
4. http://www.ibtimes.com/apple-threatens-crackdown-biggest-app-store-ranking-fraud-406764.
5. Y. Ge, H. Xiong, C. Liu, and Z.-H. Zhou. A taxi driving fraud detection system. In Proceedings of the 2011 IEEE 11th International Conference on Data Mining, ICDM '11,pages 181{190, 2011.

6. D. F. Gleich and L.-h. Lim. Rank aggregation via nuclear norm minimization. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '11, pages 60{68, 2011.

7. J. Kivinen and M. K. Warmuth.Additive versus exponentiated gradient updates for linear prediction. In Proceedings of the twenty-seventh annual ACM symposium on Theory of computing, STOC '95, pages 209{218, 1995.

8. A. Klementiev, D. Roth, and K. Small.An unsupervised learning algorithm for rank aggregation. In Proceedings of the 18th European conference on Machine Learning, ECML '07, pages 616{623, 2007.

9. A. Klementiev, D. Roth, and K. Small.Unsupervised rank aggregation with distance-based models. In Proceedings of the 25th international conference on Machine learning, ICML '08, pages 472{479, 2008.

10. E.-P. Lim, V.-A.Nguyen, N. Jindal, B. Liu, and H. W. Lauw.Detecting product review spammers using rating behaviors. In Proceedings of the 19th ACM international conference on Information and knowledge management, CIKM '10, pages 939{948, 2010.

11. A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly.Detecting spam web pages through content analysis. In Proceedings of the 15th international conference on World Wide Web, WWW '06, pages 83{92, 2006.

12. K. Shi and K. Ali.Getjar mobile application recommendations with very sparse datasets. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '12, pages 204{212, 2012.

13. N. Spirin and J. Han. Survey on web spam detection: principles and algorithms. SIGKDD Explor.Newsl., 13(2):50{64, May 2012.

14. N. Spirin and J. Han. Survey on web spam detection: principles and algorithms. SIGKDD Explor.Newsl., 13(2):50{64, May 2012.

15. Z. Wu, J. Wu, J. Cao, and D. Tao.Hysad: a semi-supervised hybrid shilling attack detector for trustworthy product recommendation. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '12, pages 985{993, 2012.

AUTHOR(S) PROFILE

Dr. K. Swathi, obtained her under-graduation in B.E., (Computer Science & Engineering) from Bharathidasan University, Trichy in 1999. She obtained her M.E. degree in Computer and Communication Engineering from Anna University, Chennai in 2004.She obtained Ph.D degree in Faculty of Information & Communication Engineering from Anna University, Chennai in 2014. Presently, she is working as Associate Professor in Computer Science & Engineering, Cauvery College of Engineering and Technology, Trichy in 2008. She has 14 years teaching experience and also she had attended many workshops, seminars and conferences on Research issues in Image processing. She has published papers in international journals and presented papers in various Conferences. She is the life member of Indian Society for Technical Education (ISTE). Her areas of interest include Image processing, Data mining, network security and Software engineering.

T. Divya, was born in Sirkazi, Tamilnadu, India, in 1991. She received the B.E. degree in Computer Science and Engineering from Loyola Institute of technology, chennai affiliated to Anna University Trichirapalli, India, in 2009, and Currently she is pursuing her M.E. degree in computer science and engineering at Cauvery Engineering college, Trichy affiliated to Anna university, Chennai., India, Her area of interest is Data mining

.

