ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

Discovery Mechanism of Ranking Fraud for Mobile Apps Applications

Sagar M. Patil[1], Somesh B. Pawar[2], Paras R. Suryawanshi[3]

Students, Dept. of Computer, Alard College of Engineering and Management, Marunje Hinjewadi, Savitribai Phule Pune University, Pune, India[123].

ABSTRACT: Nowadays everyone is using smart phone. There is need of various applications to be installed on smart phone. To download application smart phone user has to visit Apps store such as Google Play Store, Apples store etc. When user visit play store then he or she is able to see the various applications list. This list is built on the basis of promotion or advertisement. User doesn't have knowledge about the application (i.e. which applications are useful or useless). So user looks at the list and downloads the applications mostly from front page of App Store. But sometimes it happens that the downloaded application won't work or not useful. That means it is fraud in mobile application list. To avoid this fraud, we are making application in which we are going to list the applications. To list the application first we are going to find the active period of the application named as leading session. We are also investing the three types of evidences: Ranking based evidence, Rating based evidence and Review based evidence. Using these three evidences finally we are calculating aggregation of these evidences. We evaluate our application with real world data collected form play store for long time period.

KEYWORDS: Mobile Apps, ranking fraud detection, evidence aggregation, historical ranking records, rating and review.

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 leader boards. 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 "boot farms" or "human water armies" to inflate the App downloads ratings and reviews in a very short time. For example, an article from Venture Beat 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 50,000-100,000new 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.

II. RELATED WORK

This paper aims to detect users generating spam reviews or review spammers. We identify several characteristic behaviors of review spammers and model these behaviors so as to detect the spammers. In particular, we seek to model the following behaviors. First, spammers may target specific products or product groups in order to maximize their impact. Second, they tend to deviate from the other reviewer in their ratings of products. We propose scoring methods to measure the degree of spam for each reviewer and apply them on an Amazon review dataset. We then select a sub-set of highly suspicious reviewers for further scrutiny by our user evaluators with the help of a web based spammer evaluation software specially developed for user evaluation experiments. Our results show that our proposed ranking and supervised methods are effective in discovering spammer sand outperform other baseline method based on helpfulness votes alone. We finally show that the detected spammers have more significant impact on ratings compared with the unhelpful reviewers.

From this paper we have referred:-

- Concept of extracting of rating and ranking.
- Concept of extracting of review. [1]

Advances in GPS tracking technology have enabled us to install GPS tracking devices in city taxis to collect a large amount of GPS traces under operational time constraints. These GPS traces provide unparalleled opportunities for us to uncover taxi driving fraud activities. In this paper, we develop a taxi driving fraud detection system, which is able to systematically investigate taxi driving fraud. In this system, we first provide functions to find two aspects of evidences: travel route evidence and driving distance evidence. Furthermore, a third function is designed to combine the two aspects of evidences based on dempster-Shafer theory. To implement the system, we first identify interesting sites from a large amount of taxi GPS logs. Then, we propose a parameter-free method to mine the travel route evidences. Also, we introduce route mark to represent a typical driving path from an interesting site to another one. Based on route mark, we exploit a generative statistical model to characterize the distribution of driving distance and identify the driving distance evidences. Finally, we evaluate the taxi driving fraud detection system with large scale real-world taxi GPS logs. In the experiments, we uncover some regularity of driving fraud activities and investigate the motivation of drivers to commit a driving fraud by analyzing the produced taxi fraud data.

From this paper we have referred:-

- Concept of fraud detection [2]

Evaluative texts on the Web have become a valuable source of opinions on products, services, events, individuals, etc. Recently, many researchers have studied such opinion sources as product reviews, forum posts, and blogs. However, existing research has been focused on classification and summarization of opinions using natural language processing and data mining techniques. An important issue that has been neglected so far is opinion spam or trustworthiness of online opinions. In this paper, we study this issue in the context of product reviews, which are opinion rich and are widely used by consumers and product manufacturers. In the past two years, several startup companies also appeared which aggregate opinions from product reviews. It is thus high time to study spam in reviews. To the best of our knowledge, there is still no published study on this topic, although Web spam and email spam have been investigated extensively. We will see that opinion spam is quite different from Web spam and email spam, and thus requires different detection techniques. Based on the analysis of 5.8 million reviews and 2.14 million reviewers from amazon.com, we show that opinion spam in reviews is widespread. This paper analyzes such spam activities and presents some novel techniques to detect them. [3]

Many applications in information retrieval, natural language processing, data mining, and related fields require a ranking of instances with respect to specified criteria as opposed to a classification. Furthermore, for many such problems, multiple established ranking models have been well studied and it is desirable to combine their results into a joint ranking, formalism denoted as rank aggregation. This work presents a novel unsupervised learning algorithm for rank aggregation (ULARA) which returns a linear combination of the individual ranking functions based on the principle of rewarding ordering agreement between the rankers. In addition to presenting ULARA, we demonstrate its effectiveness on a data fusion task across ad hoc retrieval systems. [4]

III. EXISTING APPROACH

Generally speaking, the related works of this study can be grouped into three categories. The first category is about web ranking spam detection. Specifically, the web ranking spam refers to any deliberate actions which bring to selected web pages an unjustifiable favorable relevance or importance. Ntoulas have studied various aspects of content-based spam on the web and presented a number of heuristic methods for detecting content based spam. Zhou has studied the problem of unsupervised web ranking spam detection. Specifically, they proposed an efficient online link spam and term spam detection methods using spamicity. Recently, Spirin and Han have reported a survey on web spam detection, which comprehensively introduces the principles and algorithms in the literature. Indeed, the work of web ranking spam detection is mainly based on the analysis of ranking principles of search engines, such as Page Rank and query term frequency. This is different from ranking fraud detection for mobile Apps. The second category is focused on detecting online review spam. For example, Lim et al] have identified several representative behaviors of review spammers and model these behaviors to detect the spammers. Wu have studied the problem of detecting hybrid shilling attacks on rating data. The proposed approach is based on the semi supervised learning and can be used for trustworthy product recommendation. Xie have studied the problem of singleton review spam detection. Specifically, they solved this problem by detecting the co-anomaly patterns in multiple review based time series. Although some of above approaches can be used for anomaly detection from historical rating and review records, they are not able to extract fraud evidences for a given time period (i.e., leading session). Finally, the third category includes the studies on mobile App recommendation. For example, Yan and Chen developed a mobile App recommender system, named App joy, which is based on user's App usage records to build a preference matrix instead of using explicit user ratings. Also, to solve the sparsity problem of App usage records, Shi and Ali studied several recommendation models and proposed content based collaborative filtering model, named Eigenapp, for recommending Apps in their website Getjar. In addition, some researchers studied the problem of exploiting enriched contextual information for mobile App recommendation. For example, Zhu et al. proposed a uniform framework for personalized context-aware recommendation, which can integrate both context independency and dependency assumptions. However, to the best of our knowledge, none of previous works has studied the problem of ranking fraud detection for mobile Apps.

IV. OBJECTIVES AND SCOPE

OBJECTIVES-

1. To rank fraud for mobile application.
2. To improve the fraud detection efficiency.
3. We should first analyze the basic characteristics of leading events for extracting fraud evidences.
4. The suspicious leading events may contain very short rising and recession phases
5. We should analyze web ranking spam detection. Specifically, the web ranking spam refers to any deliberate actions which bring to selected web pages an unjustifiable favorable relevance or importance.
6. We focused on detecting online review spam.

SCOPE-

Ranking fraud in the mobile App market refers to fraudulent or deceptive activities which have a purpose of bumping up the Apps in the popularity list. Indeed, it becomes more and more frequent for App developers to use shady means, such as inflating their Apps' sales or posting phony App ratings, to commit ranking fraud. While the importance of preventing ranking fraud has been widely recognized. We provide a holistic view of ranking fraud and propose a ranking fraud detection system for mobile Apps. Specifically, we first propose to accurately locate the ranking fraud by mining the active periods, namely leading sessions, of mobile Apps. Such leading sessions can be leveraged for detecting the local anomaly instead of global anomaly of App rankings. Indeed, our careful observation reveals that mobile Apps are not always 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.

V. PROPOSED SYSTEM

First, the download information is an important signature for detecting ranking fraud, since ranking manipulation is to use so-called "bot farms" or "human water armies" to inflate the App downloads and ratings in a very short time. However, the instant download information of each mobile App is often not available for analysis. In fact, Apple and Google do not provide accurate download information on any App. Furthermore, the App developers themselves are also reluctant to release their download information for various reasons. Therefore, in this paper, we mainly focus on extracting evidences from Apps' historical ranking, rating and review records for ranking fraud detection. However, our approach is scalable for integrating other evidences if available, such as the evidences based on the download information and App developers' reputation. Second, the proposed approach can detect ranking fraud happened in Apps' historical leading sessions. However, sometime, we need to detect such ranking fraud from Apps' current ranking observations. Actually, given the current ranking ra now of an App a, we can detect ranking fraud for it in two different cases. First, if ra now > K_, where K_ is the ranking threshold introduced in Definition 1, we believe a does not involve in ranking fraud, since it is not in a leading event. Second, if ra now < K_, which means a is in a new leading event e, we treat this case as a special case that Te end ¼ te now and u2 ¼ 0. Therefore, such real-time ranking frauds also can be detected by the proposed approach.

VI. ARCHITECTURE

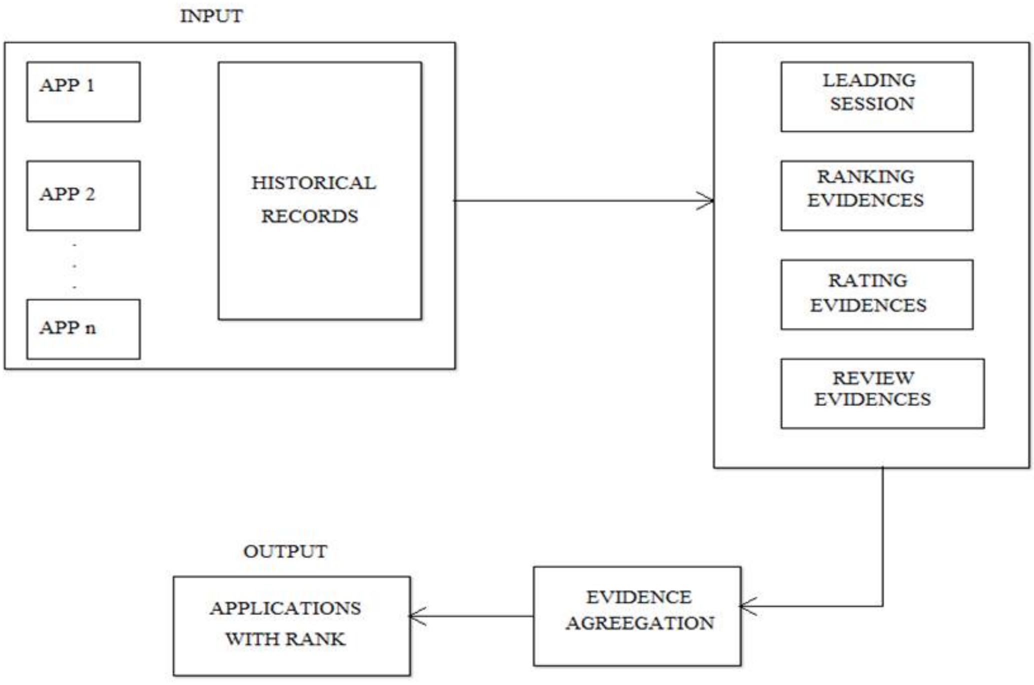


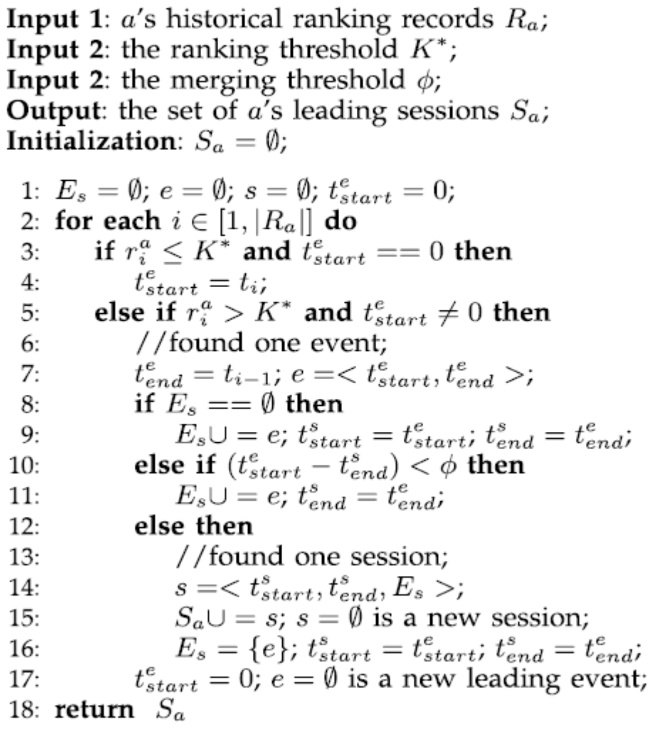
Figure 1: System Architecture

VII. ALGORITHM USED

1) Mining leading Session: -

There are two main steps for mining leading sessions. First, we need to discover leading events from the App's historical ranking records. Second, we need to merge adjacent leading events for constructing leading sessions.

The Pseudo code of mining leading sessions for a given App a:



```
Input 1: $a$'s historical ranking records $R_a$;
Input 2: the ranking threshold $K^*$;
Input 2: the merging threshold $\phi$;
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 $S_a$
```

VIII. EVENTS

Leading events

Definition 1 (Leading Event). Given a ranking threshold K e¼½testart;teend_ and corresponding rankings of a, which satisfies Ra start _ K _ a start_1 <r , and r a end _ K _ <r a endþ1 . Moreover, 8t k 2ðt e start; t e end Þ, we have r a k . Note that we apply a ranking threshold K _ K _ _ which is usually smaller than K here because K may be very big (e.g., more than 1,000), and the ranking records beyond K _ (e.g., 300) are not very useful for detecting the ranking manipulations. Furthermore, we also find that some Apps have several adjacent leading events which are close to each other and form a leading session.

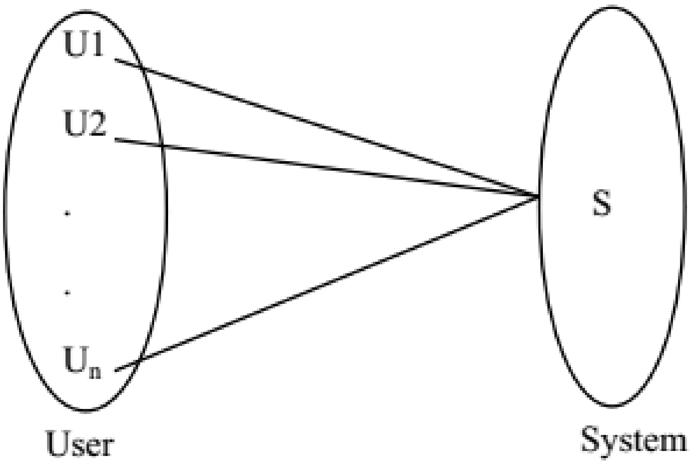
Leading Sessions

A leading session s of App a contains a time range T s ¼½t s start ;t s end _ and n adjacent leading events fe 1 ; ...;e n g, which satisfies t and there is no other leading session s s start _ ¼ t e 1 start , t that makes T _ . Meanwhile, 8i ½1; nÞ, we have ðt e iþ1 start _ t e I end s end s I ¼ t _ _ T Þ < f, where f is a predefined time threshold for merging leading events. Intuitively, the leading sessions of a mobile App represent its periods of popularity, so the ranking manipulation will only take place in these leading sessions. Therefore, the problem of detecting ranking fraud is to detect fraudulent leading sessions. Along this line, the first task is how to mine the leading sessions of a mobile App from its historical ranking records.

Identifying the Leading Sessions for Mobile APPs

There are two main steps for mining leading sessions. First, we need to discover leading events from the App's historical ranking records. Second, we need to merge adjacent leading events for constructing leading sessions. Specifically Algorithm demonstrates the pseudo code of mining leading sessions for a given App In Algorithm, we denote each leading event e and session s as tuples <t e start;t e end > and <t s start > respectively, where E is the set of leading events in session s. Specifically, we first extract individual leading event e for the given App a (i.e., Step 2 to 7) from the beginning time. For each extracted individual leading event e,we check the time span between e and the current leading session s to decide whether they belong to the same leading session based on Definition 2. Particularly, if ðt s e start; t _ t Þ < f, will be considered as a new leading session (i.e., Step 8 to 16). Thus, this algorithm can identify leading events and sessions by scanning a's historical ranking records only once.

IX. MATHEMATICAL MODEL



Many users can use one system.

Set Theory:
Our system can be represented as a set
System S = {I, O, C}
Where,
I=set of inputs
O=set of outputs
C = set of constraints
Input
Input I = {Login, Request}
Login = {Username, Password}
Request = {Search apps, search top apps, download app, Apply rating and review, Find Fraud, List apps, View History}
Users = {User, Service provider}
Username = {Username1, Username2… Username n}
Password = {Password_1, Password_2… password $_n$}
Output
Output O = {Display fraud in apps, Download start, display app list, Display history}
Constraint
C = “User should login to the system

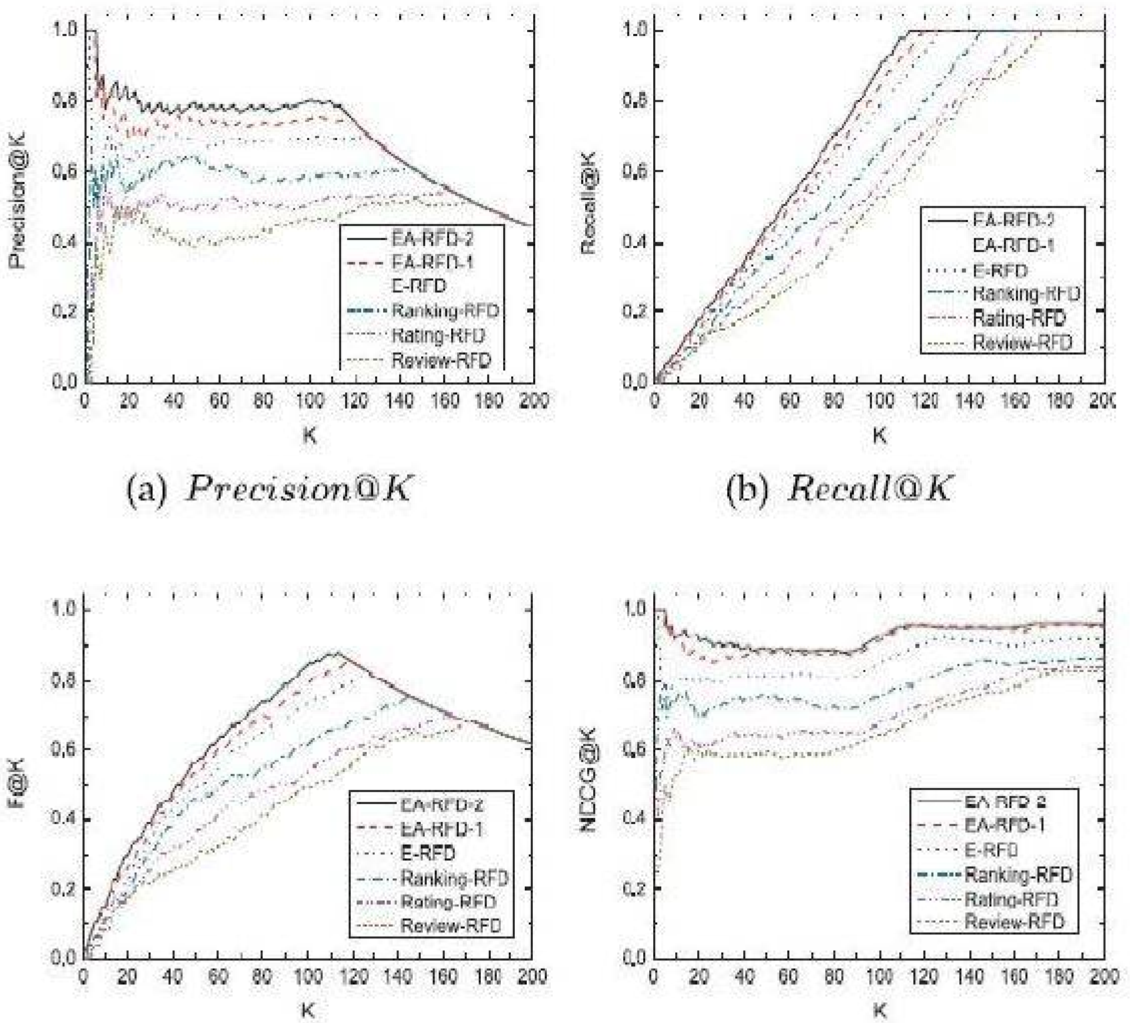
ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

EXPERIMENTAL RESULTS



(a) *Precision@K* (b) *Recall@K*

(c) *F@K* (d) *NDCG@K*

system to develop a android app that will take reviews from two different websites for single product , and analyze them with NLP for positive negative rating. In this system k-means is used to group the applications and then find out the fraud application.

X. CONCLUSION

We developed a ranking fraud detection system for mobile Apps. Specifically, we first showed that ranking fraud happened in leading sessions and provided a method for mining leading sessions for each App from its historical ranking records. Then, we identified ranking based evidences, rating based evidences and review based evidences for detecting ranking fraud. Moreover, we proposed an optimization based aggregation method to integrate all the evidences for evaluating the credibility of leading sessions from mobile Apps. An unique perspective of this approach is that all the evidences can be modeled by statistical hypothesis tests, thus it is easy to be extended with other evidences from domain knowledge to detect ranking fraud. Finally, we validate the proposed system with extensive experiments on real-world App data collected from the Androids App store.

REFERENCES

1. E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw,“Detecting product review spammers using rating behaviors,” in Proc. 19thACMInt. Conf. Inform. Knowl. Manage., 2010, pp. 939–948.
2. Y. Ge, H. Xiong, C. Liu, and Z.-H. Zhou, “A taxi driving fraud detection system,” in Proc. IEEE 11th Int. Conf. Data Mining, 2011, pp. 181–190.
3. N. Jindal and B. Liu, “Opinion spam and analysis,” in Proc. Int. Conf. Web Search Data Mining, 2008, pp. 219–230.
4. A. Klementiev, D. Roth, and K. Small, “An unsupervised learning algorithm for rank aggregation,” in Proc. 18th Eur. Conf. Mach.Learn., 2007, pp. 616–623.
5. D. F. Gleich and L.-h. Lim, “Rank aggregation via nuclear norm minimization,” in Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2011, pp. 60–68.
6. T. L. Griffiths and M. Steyvers, “Finding scientific topics,” Proc. Nat. Acad. Sci. USA, vol. 101, pp. 5228–5235, 2004.
7. L. Azzopardi, M. Girolami, and K. V. Risjbergen, “Investigating the relationship between language model perplexity and ir precision- recall measures,” in Proc. 26th Int. Conf. Res. Develop. Inform. Retrieval, 2003, pp. 369–370.
8. D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent Dirichlet allocation,” J. Mach. Learn. Res., pp. 993–1022, 2003.
9. H. Zhu, H. Xiong, Y. Ge, and E. Chen, “Ranking fraud detection for mobile apps: A holistic view,” in Proc. 22nd ACM Int. Conf. Inform. Knowl. Manage., 2013, pp. 619–628
10. S. Xie, G. Wang, S. Lin, and P. S. Yu, “Review spam detection via temporal pattern discovery,” in Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2012, pp. 823–831.
11. (2014).[Online]. Available: cohen’s_kappa
12. (2014).[Online]. Available: http://en.wikipedia.org/wiki/ information retrieval
13. (2012). [Online]. Available: http://www.lextek.com/manuals/onix/index.html
14. (2012). [Online]. Available: http://www.ling.gu.se/lager/mogul/porter-stemmer.

DOI: 10.15680/IJIRCCE.2016. 0402262