JCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Special Issue, 4, Feb 2019 E-ISSN: 2347-2693

Implementation of Data Mining Techniques to Detect Ranking Fraud

M. Mary Priyadharshini [1*], C. Premila Rosy[2]

[1]M.Sc Computer Science, Idhaya College for Women, Kumbakonam, Tamilnadu, India
[2]Department of Computer Science, Idhaya College for Women, Kumbakonam, Tamilnadu, India

Corresponding Author: premilarosy78@gmail.com

Available online at: www.ijcseonline.org

***Abstract*—** Mobile App market refers to fraudulent or deceptive activities which have a purpose of bumping up the Apps in the popularity list. It becomes more frequent for App developers to use adumbral means, such as inflating their Apps' sales or posting phony App ratings, to commit Review cheats. While the importance of preventing ranking cheat has been widely recognized, there is limited understanding and research in this area. We propose a new algorithm for this kind of the problem using Marshal Classification scam identification technique 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 find out the local anomaly instead of global anomaly of App rankings. Furthermore, we investigate three types of evidences, i.e., ranking based evidences, rating based evidences and review based evidences, by modelling Apps' ranking, valuation, review and behaviours through analytical detection principle tests using Marshal Classification Scan Analysis Technique.

***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[1]. 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 [2] App store. This is usually implemented by using so-called "bot farms" or "human water armies" to inflate the App downloads, ratings and reviews in a very short time. For example, an article from VentureBeat [3] [4] [5] [6]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 leaderboard and more than 50,000-100,000 new users could be acquired [7] [8][30] 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 [9] [10][11] in the Apple's App store. In the literature, while there are some related work, such as web ranking spam detection [21] [22], [25], [23] , online review spam detection [19], and mobile App recommendation [24] 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. Such challenge can be regarded as detecting the local anomaly instead of global anomaly of mobile Apps. 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 scalable 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, which motivates us to discover some implicit fraud patterns of mobile Apps as evidences. Indeed, our careful observation reveals that mobile Apps are not always ranked high in the leader board, 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 to identify the leading sessions of each App based on its historical ranking records[12][13]14]. 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 [15][16]17]. Nonetheless, the ranking based evidences can be affected by App developers' reputation and 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 types of fraud evidences based on Apps' rating and review history, which reflect some anomaly patterns from Apps' historical rating and review records. In addition, we develop an unsupervised evidence-aggregation method to integrate these three types of evidences for evaluating the credibility of leading sessions from mobile Apps. Fig. 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, rating and review behaviours through statistical hypotheses tests. The proposed framework is scalable and can be extended with other domaingenerated evidences for ranking fraud detection [20] [18]. Finally, we evaluate the proposed system with real-world App data collected from the Apple's App store for a long time period, i.e., more than two years. Experimental results show the effectiveness of the proposed system, the scalability of the detection algorithm as well as some regularity of ranking fraud.

II. METHODOLOGY

The proposed system has using the aggregate session collection technique has used to find the fraud mobile apps from the mobile store to download the user's correct app. It will increase the reliability of the mobile apps and it have four important stages are used to find out the fraud detection apps from the mobile store. Ranking based evidences, rating based evidences, review based evidences and evidence collection methods to use to find the apps. These three methods have to collect the information from the user according to mobile app.

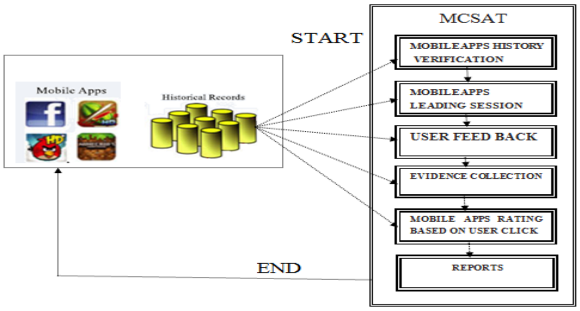
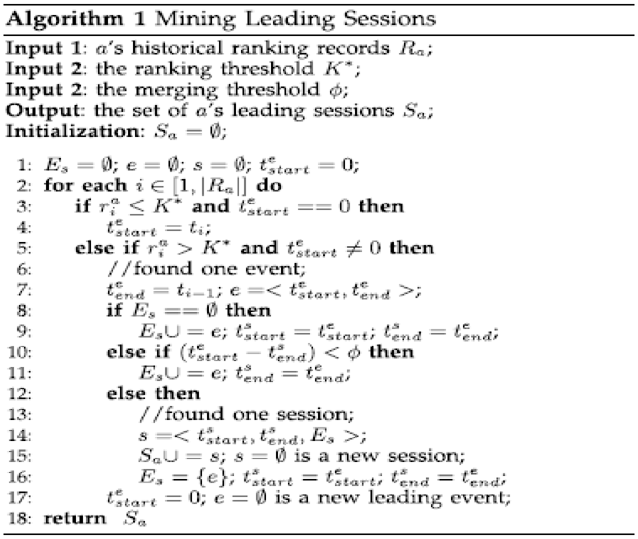

START
MCSAT
MOBILEAPPS HISTORY VERIFICATION
MOBILEAPPS LEADING SESSION
USER FEED BACK
EVIDENCE COLLECTION
MOBILE APPS RATING BASED ON USER CLICK
REPORTS
Mobile Apps
Historical Records
END

FIGURE 1: PROPOSED ARCHITECTURE

This collected thing has used to easily identify the apps correctly. This method has reduced the server storage of the fraud mobile app. We selected 50 top ranked leading sessions (i.e., most suspicious sessions), 50 middle ranked leading sessions (i.e., most uncertain sessions), and 50 bottom ranked leading sessions (i.e., most normal sessions) from each data set. Then, we merged all the selected sessions into a pool which consists 587 unique sessions from 281 unique Apps in "Top Free 300" data set, and 541 unique sessions from 213 unique Apps in "Top Paid 300" data set.



Algorithm 1 Mining Leading Sessions

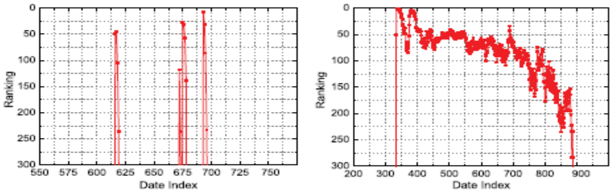
```
Input 1: $a$'s historical ranking records $R_a$;
Input 2: the ranking threshold $K^*$;
Input 2: the merging threshold $\phi$;
Output: the set of $a$'s leading sessions $S_a$;
Initialization: $S_a = \emptyset$;

1:  $E_s = \emptyset$; $e = \emptyset$; $s = \emptyset$; $t^e_{start} = 0$;
2:  for each $i \in [1, |R_a|]$ do
3:      if $r^a_i \le K^*$ 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 $S_a$
```

In Algorithm 1, we denote each leading event e and session s as tuples < te start; te end > and < ts start; ts end; Es > respectively, where Es is the set of leading events in sessions. 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 ðte start _ ts endÞ < f, e 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.

III. EXPERIMENTAL RESULTS

We can see that this App has several impulsive leading events with high ranking positions. In contrast, the ranking behaviors of a normal App's leading event may be completely different. For example, Fig. 4b shows an example of ranking records from a popular 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.

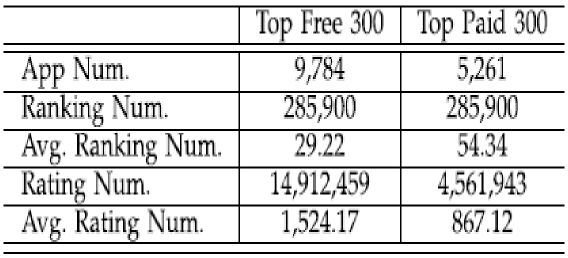


(a) Example 1 (b) Example 2

FIGURE 2: TWO REAL-WORLD EXAMPLES OF LEADING EVENTS

In fact, once a normal App is ranked high in the leaderboard, it often owns lots of honest fans and may attract more and more users to download. Therefore, this App will be ranked high in the leaderboard for a long time. Based on the above discussion, we propose some ranking based signatures of leading sessions to construct fraud evidences for ranking fraud detection.

TABLE 1 Statistics of the Experimental Data



	Top Free 300	Top Paid 300
App Num.	9,784	5,261
Ranking Num.	285,900	285,900
Avg. Ranking Num.	29.22	54.34
Rating Num.	14,912,459	4,561,943
Avg. Rating Num.	1,524.17	867.12

IV. CONCLUSION

Developing and discovering a ranking based fraud detection system for mobile Apps Successfully. 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. In the future, Plan to investigate more effective fraud evidences and analyze the latent relationship among rating, review and rankings based hypothesis test for more than one mobile App. Moreover, we will extend our ranking fraud detection approach with other mobile App related services, such as mobile Apps recommendation, and also for enhancing user experience.

REFERENCES

[1](2014).[Online]. Available: http://en.wikipedia.org/wiki/cohen's_kappa

[2](2014). [Online]. Available: http://en.wikipedia.org/wiki/information_retrieval.

[3] (2012). [Online]. Available: https://developer.apple.com/news/index.php?id=02062012a

[4] (2012). [Online]. Available: http://venturebeat.com/2012/07/03/apples-crackdown-on-app-ranking-manipulation/

[5] (2012). [Online]. Available: http://www.ibtimes.com/applethreatens-crackdown-biggest-app-store-ranking-fra ud-406764

[6] (2012). [Online]. Available: http://www.lextek.com/manuals/onix/index.html

[7] (2012). [Online]. Available: http://www.ling.gu.se/lager/mogul/porter-stemmer.

[8] 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.

[9] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," J. Mach. Learn. Res., pp. 993–1022, 2003.

[10] 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.

[11] 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.

[12] T. L. Griffiths and M. Steyvers, "Finding scientific topics," Proc. Nat. Acad. Sci. USA, vol. 101, pp. 5228–5235, 2004.

[13] G. Heinrich, Parameter estimation for text analysis, " Univ. Leipzig, Leipzig, Germany, Tech. Rep., http://faculty.cs.byu.edu/~ringger/CS601R/papers/Heinrich-GibbsLDA.pdf, 2008.

[14] N. Jindal and B. Liu, "Opinion spam and analysis," in Proc. Int. Conf. Web Search Data Mining, 2008, pp. 219–230.

[15] J. Kivinen and M. K. Warmuth, "Additive versus exponentiated gradient updates for linear prediction," in Proc. 27th Annu. ACM Symp. Theory Comput., 1995, pp. 209–218.

[16] A. Klementiev, D. Roth, and K. Small, "An unsupervised learning.

[17] A. Klementiev, D. Roth, and K. Small, "Unsupervised rank aggregation with distance-based models," in Proc. 25th Int. Conf. Mach. Learn., 2008, pp. 472–479.

[18] A. Klementiev, D. Roth, K. Small, and I. Titov, "Unsupervised rank aggregation with domain-specific expertise," in Proc. 21st Int. Joint Conf. Artif. Intell., 2009, pp. 1101–1106.

[19] 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.

[20] Y.-T. Liu, T.-Y. Liu, T. Qin, Z.-M. Ma, and H. Li, "Supervised rank aggregation," in Proc. 16th Int. Conf. World Wide Web, 2007, pp. 481–490.

[21] A. Mukherjee, A. Kumar, B. Liu, J. Wang, M. Hsu, M. Castellanos, and R. Ghosh, "Spotting opinion spammers using behavioral footprints," in Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2013, pp. 632–640.

[22] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly, "Detecting spam web pages through content analysis," in Proc. 15th Int. Conf. World Wide Web, 2006, pp. 83–92.

[23] G. Shafer, A Mathematical Theory of Evidence. Princeton, NJ, USA: Princeton Univ. Press, 1976.

[24] K. Shi and K. Ali, "Getjar mobile application recommendations with very sparse datasets," in Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2012, pp. 204–212.

[25] N. Spirin and J. Han, "Survey on web spam detection: Principles and algorithms," SIGKDD Explor. Newslett., vol. 13, no. 2, pp. 50–64, May 2012.