ISSN (Online) : 2454-4159
International Journal of Advanced Research in Science Management and Technology
Volume 2, Issue 6, June 2016

Detection of fraud and prevention of mobile apps using aggregation method

Vivek Pingale[1], Laxman Kuhile[2], Pratik Phapale[3], Pratik Sapkal[4], Prof. Swati jaiswal[5]

UG Student, Dept. Of Computer Engg, SKNSITS, Lonavala, Pune, M.S., India[1,2,3,4]
Professor, Dept. Of Computer Engg., , SKNSITS, Lonavala, Pune, M.S., India [5]

ABSTRACT— The number of mobile Apps has grown at a huge rate over the past few years. Ranking fraud in the mobile App market refers to fraudulent or fake activities which have a purpose of strike up the Apps in the popularity list. It becomes more and more frequent for App developers to posting bogus App ratings, to commit ranking fraud. While the importance of preventing ranking fraud has been widely recognized, there is limited understanding and research in this area. To this end, in this paper, we provide a brief 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 by using mining leading session algorithm. Furthermore, we investigate three types of evidences, i.e., ranking based evidences, rating based evidences and review based evidences, by studying historical records. we used an optimal aggregation method to integrate all the evidences for fraud detection. Finally, we evaluate the proposed system with real-world App data collected from the Google App Store for a long time period. In the experiments, we validate the effectiveness of the proposed system, and show the scalability of the detection algorithm as well as some regularity of ranking fraud activities.

KEYWORDS – Mobile Apps, Ranking Fraud Detection, Evidence Aggregation, Historical Ranking Records, Rating and Review.

I. INTRODUCTION

Nowadays billions of mobile apps releasing at huge rate.For example, as of the end Of 2015, there are more than 25 million Apps at Google Play. To stimulate the development of mobile Apps, many App stores launched daily App leaderboards, which demonstrate the chart rankings of most popular Apps. Indeed, the App leaderboard is one of the most important ways for promoting mobile Apps. A higher rank on the leaderboard usually leads to a huge number of downloads and million dollars in revenue. Therefore, App developers tend to explore various ways such as advertisement 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 , some App developers have a intention of creating unuseful apps for some fraudulent means for their advantage and manually updates a ranking chart as their wish.

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. In this paper, we provide a brief 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 by using mining leading session algorithm. Such leading sessions can be useful for detecting 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 modeling Apps' ranking, rating and review behaviors by analyzing its historical records. we propose an optimization based aggregation method to integrate all the evidences for fraud detection.

ISSN (Online) : 2454-4159
International Journal of Advanced Research in Science Management and Technology
Volume 2, Issue 6, June 2016

II. PROPOSED SYSTEM

In proposed system we overcome the drawbacks of Mining leading session algorithm which is based on ranking, review & rating. 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 download and ratings in a very short time. However, the instant download information of each mob. 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, the focus is 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 evidences based on the download information and App developers' reputation. Second, the proposed approach can detect ranking fraud happened in A,' historical leading sessions.

Ranking fraud detection in mobile apps is actually to detect ranking fraud within leading session of mobile apps. Specifically we identified first leading sessions based on Apps historical ranking records. Then with the analysis of Apps' ranking behaviours we characterized some fraud evidences from historical records.

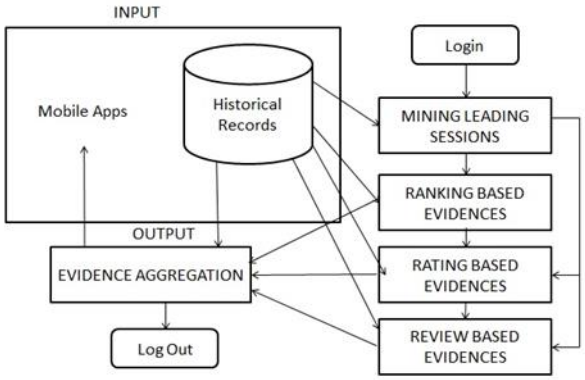

Fig. 1 System architecture

The ranking based evidences can be affected some Apps' developer reputation and some legitimate marketing campaigns, such as "limited-time discount". This method is not enough to detect fraudulent Apps' so we propose two new methods of fraud evidences based on Apps' historical rating and review records. Additionally, we developed an unsupervised evidence-aggregation method to integrate these types of evidences.

Extracting Evidence For Ranking Fraud Detection

In this section, we study how to extract and combine fraud evidences for ranking fraud detection.

EVIDENCE AGGREGATION ALGORITHM

1. Analyze the historical records of mobile apps.
2. Differentiate the evidences as Ranking based, Rating based, Review based.
3. Aggregate these evidences by using optimal aggregation algorithm.
4. Design Android application framework

ISSN (Online) : 2454-4159
International Journal of Advanced Research in Science Management and Technology
Volume 2, Issue 6, June 2016

Fagin's algorithm [17]

In this section, They discuss FA (Fagin's Algorithm) [Fag99]. This algorithm is implemented in Garlic [CHS+95], an experimental IBM middleware system; see [WHRB99] for interesting details about the implementation and performance in practice. Chaudhuri and Gravano [CG96] consider ways to simulate FA by using ''filter conditions'', which might say, for example, that the color score is at least 0.2.6 FA works as follows.

1. Do sorted access in parallel to each of the m sorted lists Li: (By ''in parallel'', we mean that we access the top member of each of the lists under sorted access, then we access the second member of eachof the lists, and so on.)7 Wait until there are at least k ''matches'', that is, wait5QBIC is a trademark of IBM Corporation.6Chaudhuri and Gravano originally saw an early version of the conference paper (in the 1996 ACM Symposium on Principles of Database Systems) that expanded into the journal version [Fag99]. It is not actually important that the lists be accessed ''in lockstep''. In practice, it may be convenient to allow the sorted lists to be accessed at different rates, in batches, etc. Each of the algorithms in this paper where there is ''sorted access in parallel'' remain correct even when sorted access is not in lockstep. Furthermore, all of our instance optimality results continue to hold even when sorted access is not in lockstep, as long as the rates of sorted access of the lists are within constant multiples of each other.

R. Fagin et al. / Journal of Computer and System Sciences 66 (2003) 614–656 619until there is a set of at least k objects such that each of these objects has been seen in each of the m lists.

2. For each object R that has been seen, do random access as needed to each of the lists Li to find the ith field xi of R:

3. Compute the grade tðRÞ ¼ tðx1;y; xmÞ for each object R that has been seen. Let Y be a set containing the k objects that have been seen with the highest grades (ties are broken arbitrarily). The output is then the graded set fðR; tðRÞÞ j RAYg:It is fairly easy to show [Fag99] that this algorithm is correct for monotone aggregation functions t (that is, that the algorithm successfully finds the top k answers). If there are N objects in the database, and if the orderings in the sorted lists are probabilistically independent, then the middleware cost of FA is OðNðm 1Þ=mk1=mÞ; with arbitrarily high probability [Fag99].

An aggregation function t is strict [Fag99] if tðx1;y; xmÞ ¼ 1 holds precisely when xi ¼ 1 for every i: Thus, an aggregation function is strict if it takes on the maximal value of 1 precisely when each argument takes on this maximal value. We would certainly expect an aggregation function representing the conjunction to be strict (see the discussion in [Fag99]). In fact, it is reasonable to think of strictness as being a key characterizing feature of the conjunction Fagin shows that his algorithm is optimal with high probability in the worst case if the aggregation function is strict (so that, intuitively, we are dealing with a notion of conjunction),and if the orderings in the sorted lists are probabilistically independent. In fact, the access pattern of FA is oblivious to the choice of aggregation function, and so for each fixed database, the middleware cost of FA is exactly the same no matter what the aggregation function is. This is true even for a constant aggregation function; in this case, of course, there is a trivial algorithm that gives us the top k answers (any k objects will do) with Oð1Þ middleware cost. So FA is not optimal in any sense for some monotone aggregation functions t: As a more interesting example, when the aggregation function is max (which is not strict), it is shown in [Fag99] that there is a simple algorithm that makes at most mk sorted accesses and no random accesses that finds the top k answers. By contrast, as we shall see, the algorithm TA is instance optimal for every monotone aggregation function, under very weak assumptions

ISSN (Online) : 2454-4159
International Journal of Advanced Research in Science Management and Technology
Volume 2, Issue 6, June 2016

III. RESULTS AND DISCUSSION

User Login Form

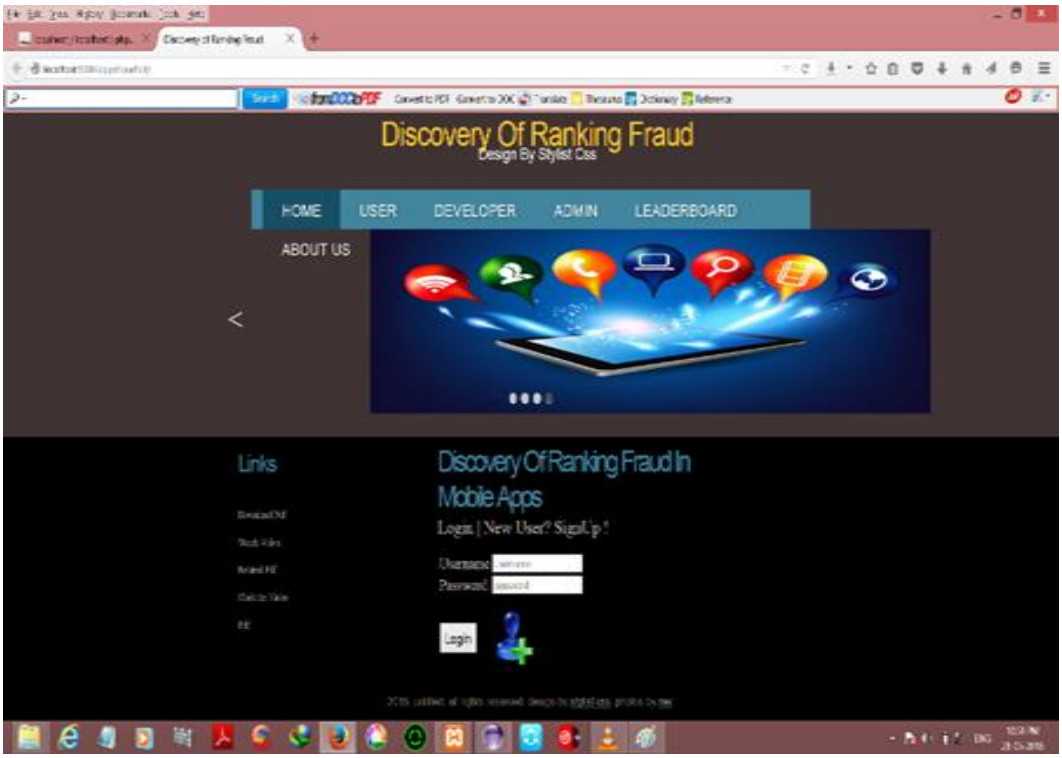

Fig. 2 login form

User uses user Login form which gives access to user domain. It provide different options to new users as Register, create form, sign up , login. To login user have to enter username and password. Only authorized users can be access. If user is not authorized then he need to sign up and create his account.

Ranking Based Evidences

We should first analyze the basic characteristics of leading events for extracting fraud evidences. Therefore, we should first analyze the basic characteristics of leading events for extracting fraud evidences.

1. By analysing the Apps' historical ranking records, we observe that Apps' ranking behaviors in a leading eventual ways satisfy a specific ranking pattern, which consists of three different ranking phases, namely

- Rising phase:
- Maintaining phase:
- Recession phase:

2. In each leading event, an App's ranking first increases to a peak position in the leader board (i.e., rising phase), then keeps such peak Position for a period (i.e., maintaining phase), and finally decreases till the end of the event.

Rating Based Evidences:

The ranking based evidences are useful for ranking fraud detection.

ISSN (Online) : 2454-4159
International Journal of Advanced Research in Science Management and Technology
Volume 2, Issue 6, June 2016

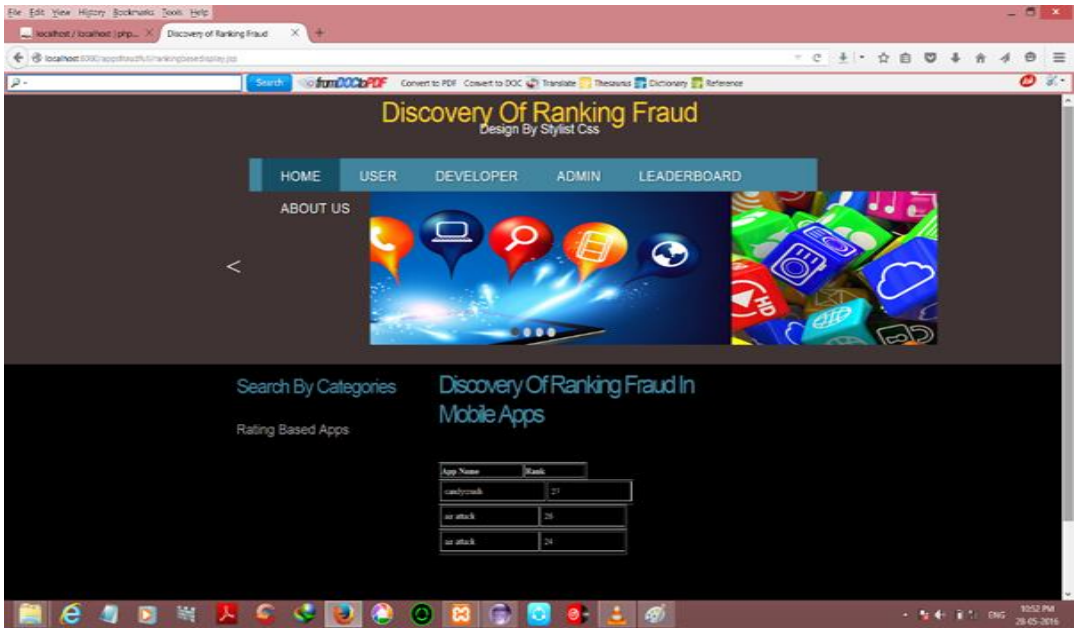

Fig. 3 Rating Based Evidences

Review Based Evidences:

Most of the App stores also allow users to write some textual comments as App reviews. Such review scan reflect the personal perceptions and usage experiences of existing users for particular mobile Apps.

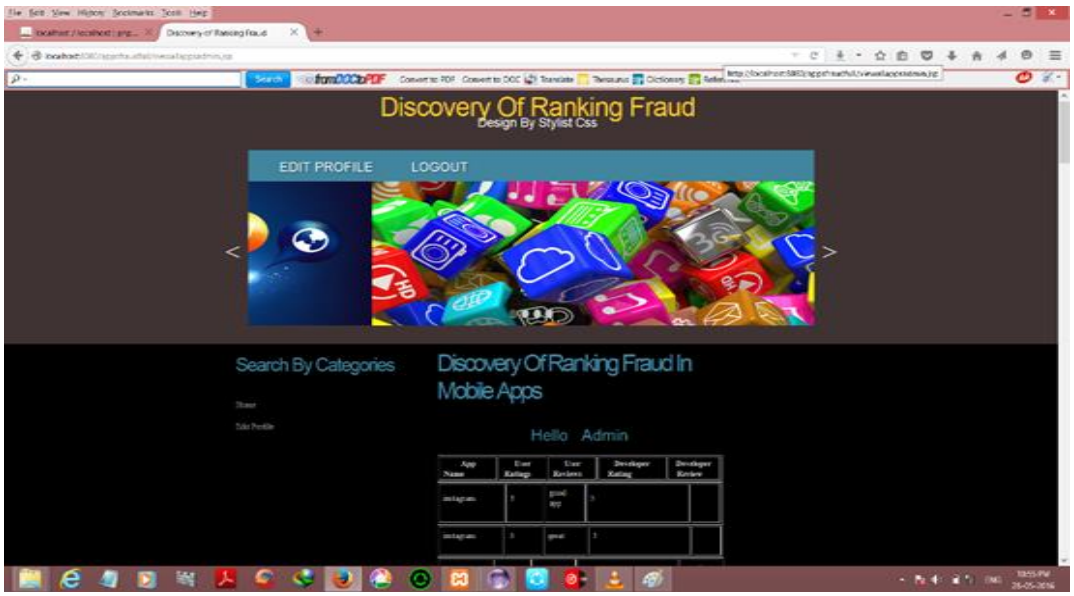

Fig. 4 Admin View

IV. RELATED WORK

The origin of research on rank aggregation can be traced back to the eighteenth century, when it was studied in social choice theory and applied into political elections. In recent years, rank aggregation gets spotlight again in many new applications, such as genome database construction, document filtering, database middleware construction, spam webpage detection, meta-search, word association finding, multiple search, and similarity search. There are two types of rank aggregation: score-based and order based. In the former the aggregation function takes score information from

the individual base rankers as input, while in the latter it only utilizes order information. Order-based aggregation fits well with meta-search, as in meta-search only order information from base rankers is available; this is also the main focus of the research in this paper. Existing methods for order-based aggregation includes, for example, Borda Count , median rank aggregation , genetic algorithm , fuzzy logic based rank aggregation method and Markov Chain based rank aggregation .

Borda Count ranks entities based on their positions in the ranking lists. For example, the entities are sorted according to the number of entities that are ranked below them in all the ranking lists. Median rank aggregation sorts the entities based on the medians of their ranks in all the ranking lists. Markov Chain based rank aggregation assumes that there exists a Markov Chain on the entities and the order relations between entities in the ranking lists represents the transitions in Markov Chain. The stationary distribution of the Markov Chain is utilized to rank the entities. Dwork et al proposed four methods (denoted as MC1, MC2, MC3, and MC4) to construct the transition probability matrix of the Markov Chain. The unsupervised methods described above implicitly conduct majority voting in their final ranking decisions. That is to say, these methods treat all the ranking lists equally and give high ranks to those entities ranked high by most of the rankers. This assumption may not hold in practice, however. For example, in meta-search, ranking lists are generated by different search engines with different capacities and accuracies. It is not reasonable to treat the results of the search engines equally.

V. CONCLUSION

In this paper, 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 a mining Leading session algorithm for obtain mining leading session and aggregation method. In the future, we plan to study more effective fraud evidences and analyze the latent relationship among rating, review and rankings. Moreover, we will extend our ranking fraud detection approach with other mobile App related services, such as mobile Apps recommendation, for enhancing user experience.

REFERENCES

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

[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] **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.**

[4] **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.**

[5] **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**

[6] **Hengshu Zhu, Hui Xiong, Senior Member, IEEE, Yong Ge, and Enhong Chen, Senior Member, IEEE, "Discovery of Ranking Fraud for Mobile Apps", vol.13,n0.1,Jan 2015**

[7] **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.**

[8] **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.**

[9] **B. Zhou, J. Pei, and Z. Tang, "A spamicity approach to web spam detection," in Proc. SIAM Int. Conf. Data Mining, 2008, pp. 277–288.**

ISSN (Online) : 2454-4159
International Journal of Advanced Research in Science Management and Technology
Volume 2, Issue 6, June 2016

[10] 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

[11] Z. Wu, J. Wu, J. Cao, and D. Tao, "HySAD: A semi-supervised hybrid shilling attack detector for trustworthy product recommendation," in Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2012, pp. 985–993.

[12] 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.

[13] 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.

[14] B. Yan and G. Chen, "AppJoy: Personalized mobile application discovery," in Proc. 9th Int. Conf. Mobile Syst., Appl., Serv., 2011, pp. 113–126.

[15] H. Zhu, H. Cao, E. Chen, H. Xiong, and J. Tian, "Exploiting enriched contextual information for mobile app classification," in Proc. 21st ACMInt. Conf. Inform. Knowl. Manage., 2012, pp. 1617–1621.

[16] H. Zhu, E. Chen, K. Yu, H. Cao, H. Xiong, and J. Tian, "Mining personal context-aware preferences for mobile users," in Proc. IEEE 12th Int. Conf. Data Mining, 2012, pp. 1212–1217.

[17] Ronald Fagin,a,_ Amnon Lotem,b and Moni Naorc,1, "Optimal aggregation algorithms for middleware"Journal of Computer and System Sciences 66 (2003) 614–656 ,1 April 2002