

# Recognition of Ranking Frauds in Mobile Apps

Abhiilash T P<sup>1</sup>, L Dinesha<sup>2</sup>P.G. Student, Department of Computer Science & Engineering, SSIT, Tumakuru, Karnataka, India<sup>1</sup>Assistant Professor, Department of Computer Science & Engineering, SSIT, Tumakuru, Karnataka, India<sup>2</sup>

**ABSTRACT:** There are millions of apps available in market for the application of mobile users. However, all the mobile users first prefer high ranked apps when downloading it. But we cannot guarantee the reliability for the downloaded application since there is increasing number of ranking frauds. 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, there is limited understanding and research in this area. To this end, in this paper, 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. 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 through statistical hypotheses tests. In addition, we propose an optimization based aggregation method to integrate all the evidences for fraud detection. Finally, we evaluate the proposed system with real-world App data collected from the OS 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. There are in huge number of official and unofficial markets available for mobile users to get variety of application. However, we cannot guarantee that the applications available in the market are trust worthy. Therefore, the application needs to be validated. In this paper we are introducing new protocol for detecting malicious apps.

**KEYWORDS:** Rank Based evidence, Ranking Fraud, Mobile Applications.

## I. INTRODUCTION

Mobile telephone fraud is the unauthorized use of the telecommunications network accomplished via deception. Mobile telephone fraud is a tremendous difficulty for network vendors and their customers: in some local areas it is estimated that more than half the use is fraudulent.

To stimulate the progress of cell Apps, many App outlets launched daily App leader boards, which demonstrate the chart rankings of most trendy Apps. Certainly, the App leader board is likely one of the primary ways for promoting cellular Apps. A bigger rank on the leader board normally leads to a tremendous quantity of downloads and million bucks in earnings. Consequently, App builders are likely to explore various approaches corresponding to promoting campaigns to promote their Apps in an effort to have their Apps ranked as high as possible in such App leader boards.

In the literature, even as there are some associated work, similar to web ranking junk mail detection [1], [2] online overview junk mail detection and mobile App recommendation the main issue of detecting ranking fraud for cellular Apps is still underexplored.

We propose to advance a ranking fraud detection process for mobile Apps. Along this line, we establish a couple of main challenges. First, ranking fraud does not perpetually happen in the whole life cycle of an App, so we need to discover the time when fraud occurs. Such challenge can be regarded as detecting the local anomaly instead of global anomaly of cell Apps. 2nd, as a result of the big quantity of mobile Apps, it's intricate to manually label ranking fraud for

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

each and every App, so it's fundamental to have a scalable method to mechanically detect ranking fraud without making use of any benchmark information. In the end, due to the dynamic nature of chart rankings, it's not easy to establish and confirm the evidences linked to ranking fraud, which motivates us to observe some implicit fraud patterns of cellular Apps as evidences.

Indeed, our careful commentary displays that cell Apps aren't constantly ranked high within the leader board, however handiest in some leading events, which type distinct leading Sessions.

Ranking fraud most of the time happens in these leading sessions. Therefore, detecting rating fraud of cellular Apps is truly to detect ranking fraud inside leading sessions of cellular Apps. Specifically, we first recommend a easy but powerful algorithm to identify the leading sessions of each and every App established on its historical rating files. Then, with the analysis of Apps' ranking behaviors, we discover that the fraudulent Apps quite often have exclusive ranking patterns in each and every main session when compared with normal Apps. Therefore, we symbolize some fraud evidences from Apps' historical ranking documents, and enhance three services to extract such ranking based fraud evidences. Nonetheless, the rating based evidences can be affected through App builders' fame and some reliable advertising campaigns, reminiscent of "constrained-time discount". As a result, it is not ample to only use ranking established evidences. Consequently, we further suggest two forms of fraud evidences headquartered on Apps' ranking and assessment historical past, which reflect some anomaly patterns from Apps' ancient rating and evaluation documents. Furthermore, we boost an unsupervised evidence-aggregation system to integrate these three types of evidences for evaluating the credibility of main periods from mobile Apps detection system for mobile Apps.

## II. RELATED WORK

1: K. Shi and K. Ali, "Getjar mobile application recommendations with very sparse datasets", International Conference on Knowledge Discovery and Data Mining, 2012.

In this paper compared a latent factor (PureSVD) and a memory-based model with our novel PCA-based model, which we call Eigenapp. Here used both accuracy and variety as evaluation metrics. PureSVD did not perform well due to its reliance on explicit feedback such as ratings, which we do not have. Memory-based approaches that perform vector operations in the original high dimensional space over predict popular apps because they fail to capture the neighbourhood of less popular apps. They have high accuracy due to the concentration of mass in the head, but did poorly in terms of variety of apps exposed. Eigenapp, which exploits neighbourhood information in low dimensional spaces, did well both on precision and variety, underscoring the importance of dimensionality reduction to form quality neighbourhoods in high kurtosis distributions.

2: N. Spirin and J. Han, "Survey on web spam detection: principles and algorithms", SIGKDD, 2012.

In this paper presented a systematic review of web spam detection techniques with the focus on algorithms and underlying principles. Here 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, author performed a sub categorization of link-based category into five groups based on ideas and principles used: labels propagation, link pruning and reweighting, labels refinement, graph regularization, and feature based. Here also define the concept of web spam numerically and provide a brief survey on various spam forms. Finally, summarized the observations and underlying principles applied for web spam detection.

## III. PROPOSED SYSTEM

Figure 1 indicates the block diagram of proposed architecture. For locating the leading session we'd like the ancient data. Old files accrued from quite a lot of sources. Then discovering the leading session shall be finished. This can be completed with the aid of discovering leading events from the App's historic rating documents. Second, we need to merge adjoining main pursuits for setting up leading periods.

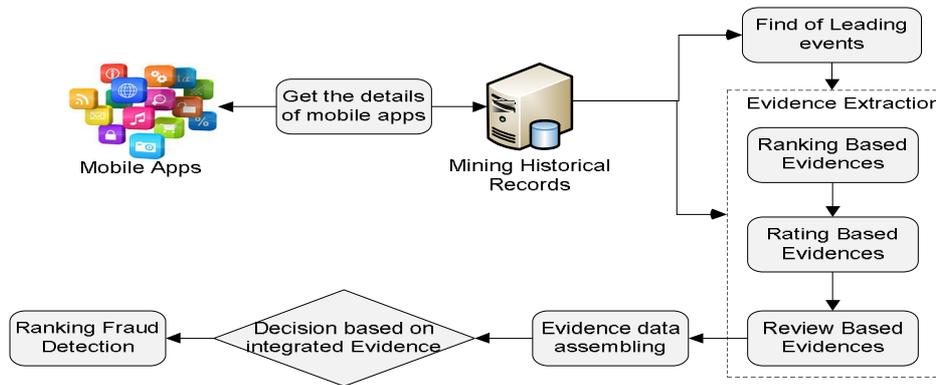


Figure 1: Proposed architecture

In the next step we ought to collect the evidences from the historic documents. Evidences will likely be collected headquartered on prior ranking, rating and experiences respectively.

### c. Extraction of Evidences

In determining ranking frauds historical evidences plays vital role. Here also we collect the useful evidences based on rating, ranking and experiences.

#### C1.1. Ranking Based Evidences

By way of inspecting the Apps' ancient ranking files, we realize that Apps' rating behaviors in a leading event continuously fulfil a targeted rating pattern, which contains three specific ranking phases, namely, rising phase, maintaining phase and recession phase. Especially, in every leading event, an App's ranking first increases to a top position in the leader board (i.e., rising phase), then keeps such top position for a period (i.e., maintaining phase), and ultimately decreases till the tip of the event (i.e., recession phase).

As soon as a common App is ranked high within the leader board, it obviously owns lots of honest enthusiasts and would appeal to more and more purchasers to download. Consequently, this App is usually ranked high inside the leader board for a long time. From the above discussion, we propose some ranking established evidences of major sessions to assemble fraud evidences for ranking fraud detection.

#### Proof 1:

We use two shape parameters  $\theta_1$  and  $\theta_2$  to quantify the score patterns of the rising phase and the recession phase of App a's main event e, which may also be computed by using

$$\theta_1^e = \arctan \frac{n(K-r_b^a)x^2}{t_b^e - t_a^e} \quad (1)$$

$$\theta_2^e = \arctan \frac{(K-r_c^e)}{t_a^e - t_c^e} \quad (2)$$

Where, k is the rating threshold Intuitively, a higher  $\theta_1$  may just point out that the App has been bumped to a high rank within a short interval, and a tremendous  $\theta_2$  could factor out that the App has dropped from a excessive rank to the bottom within a brief interval. Hence, a leading session, which has more leading movements with significant  $\theta_1$  and  $\theta_2$  values, has larger chance of getting rating fraud. Correct now, we outline a fraud signature  $\theta_s$  for a predominant session as follows

$$\theta_s = \frac{1}{|E_s|} \sum_{e \in E_s} \theta_1^e + \theta_2^e \quad (3)$$

Where  $|E_s|$  quantity of primary routine in session s. Intuitively, if a leading session s includes significantly higher  $\theta_s$  in evaluation with exceptional leading durations of Apps within the leader board, it has high hazard of having rating fraud. To grab this, we propose to apply statistical speculation scan for computing the value of  $\theta_s$  for each major session. In designated, we define two statistical hypotheses as follows and compute the p-price of every leading session.

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

- The signature  $\theta_s$  of leading session  $s$  is no longer useful for detecting ranking fraud.
- The signature  $\theta_s$  of main session  $s$  is enormously better than expectation.

### C1.2 Experience based evidences:

Experiences are very predominant proof in deciding whether the data is riskless or no longer. However most likely it's difficult to gauge established on simplest experiences.

Experiences can replicate the individual perceptions and utilization experiences of existing users for targeted cellular Apps. Most likely, comparison manipulation is no doubt probably the most primary perspective of App rating fraud. Mainly, earlier than downloading or purchasing a brand new cell App, users more often than not first of all learns its prior experiences to ease their resolution making, and a cell App entails more positive reports may just attract further customers to download. For that reason, imposters often put up false reviews inside the leading classes of a targeted App so that you could inflate the App downloads, and thus propel the App's ranking positions within the leader board.

Proper here we endorse two fraud evidences based on App's stories behaviors in main sessions for detecting rating fraud.

### Proof 1:

Many of the experience manipulations are applied by way of boot farms when you consider that of the immoderate expense of human useful resource. As a consequence, review spammers most of the time submits a couple of duplicate or near-reproduction studies on the equal App to inflate download. In difference, the traditional App perpetually has various studies given that customers have exceptional individual perceptions and utilization. From the above observations, right here we define a fraud signature  $Sim(s)$ , which denotes the natural mutual similarity between the reports inside main session  $s$ . In particular; this fraud signature can be computed with the help of following steps.

- 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  $\rightarrow$  play", "better  $\rightarrow$  good").
- We build a normalized words vector  $\vec{w_c} = dim(n)$  for each review  $c$ , where  $n$  indicates the number of normalized words in all reviews of  $s$ .

$$dim[i] = \frac{freq_{i,c}}{\sum_i freq_{i,c}} \quad (1 \leq i \leq n) \quad (4)$$

Freq is the frequency of  $ith$  word in  $c$

Finally, we can calculate the similarity between two reviews  $c_i$  and  $c_j$  by the Cosine similarity  $COS(W_{c_i}, W_{c_j})$ . Thus the fraud signature  $Sim(s)$  is

$$Sim(s) = 2 * \sum_{1 \leq i < j < N_s} COS(W_{c_i}, W_{c_j}) \setminus N_s * (N_s - 1) \quad (5)$$

Where  $N_s$  is the number of stories for the period of principal session  $s$ . Intuitively, the better valued at of  $sim(s)$  suggests further reproduction/close-reproduction reviews in  $s$ . For that reason, if a leading session has significantly bigger worth of  $Sim(s)$  in comparison with one of kind fundamental courses of Apps in the chief board, it has excessive likelihood of having ranking fraud.

To compute this, we define statistical hypotheses to compute the value of  $Sim(s)$  for each main session as follows.

- The signature  $Sim(s)$  of leading session  $s$  is not useful for detecting ranking fraud.
- The signature  $Sim(s)$  of leading session  $s$  is significantly higher than expectation.

Here, we use the Gaussian approximation to compute the p-worth with the above hypotheses. Above all, we count on  $Sim(s)$  follows the Gaussian distribution,

$$\varphi_6^{(s)} = p(\mu_{sim}, \sigma_{sim}) \geq sim(s) \quad (6)$$

### d. Proof Aggregation:

After extracting three forms of fraud evidences, the following assignment is the way to mix them for ranking fraud detection. Certainly, there are various rating and proof aggregation approaches in the literature, comparable to

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

permutation established units [7], rating centered items [1], [6] and Dumpster-Shafer principles [1], [2]. However, some of those approaches focus on finding out a worldwide ranking for all candidates. This isn't right for detecting rating fraud for brand new Apps. Different ways are situated on supervised learning strategies, which depend upon the labeled training data and are tough to be exploited. Instead, we advocate an unmonitored method centered on fraud similarity to combine these evidences.

Certainly, we outline the final proof ranking  $\Psi(s)$  as a linear combination of all of the existing evidences as observe that, right here we advise to use the linear blend since it has been validated to be robust and is largely used in imperative domains, such as rating aggregation [6], [8].

$$\varphi(s) = \sum_{i=1}^N w_i * \varphi_i(s), s.t. \sum_{i=1}^N w_i = 1 \quad (7)$$

d. Algorithm for mining leading sessions

## IV. EXPECTED RESULTS



Figure 2: Home Page

Home page is the first page where the user will be provided with the information about the app store and will it work important point about the app store. From this page the links will be given to the user page , service provider and administrator pages by linking the links to the required page can get .

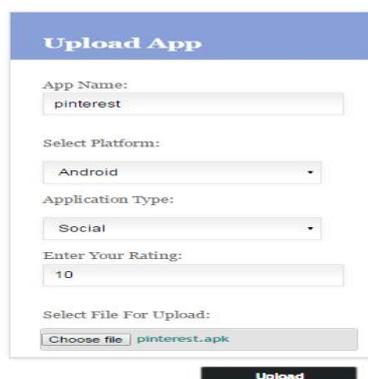


Figure 3: Service Provider Uploading Application

## International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

The Service Provider can upload the application to app store by using the Service Provider page, While uploading the application he has to give certain details like platform of the application type of the application an by choosing the file he can upload



Figure 4: Application List

The users will be provided with the list of application which have present in the app store by which the user can select or search the application which he requires with the list each application will be having the type of the application with its name to get the more information can be get by clicking on it .



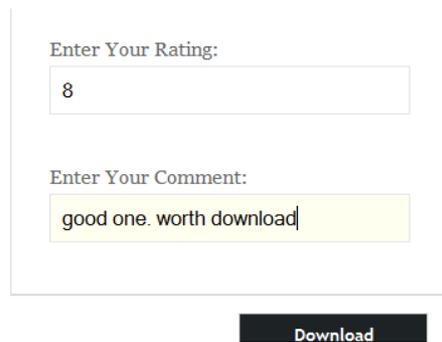
Figure 5: User Rating and number of downloads

The user will be provided with the more information about the each application like what is the users rating about the application and numbers of application has be downloaded till now by the users though the app store

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016



Enter Your Rating:  
8

Enter Your Comment:  
good one. worth download

Download

Figure 6: User Comments and Rating when downloading

After downloading the application the users will be provided with the users to give rating about the application and , The user can also enter the comments about the application how that application works was it useful which can be use to evaluate that application

Application	Pinterest
<b>Fraud Percentage Based on Rating</b>	20%
<b>Fraud Percentage Based on Review</b>	15%
<b>Fraud Percentage Based on Ranking</b>	22%
<b>Ranking Result</b>	8

Table 1: Fraud Application Ranking Detection Result

Fraud Application Ranking Detection Result table after evaluation of the application based on the collected data the result will giving in the percentage of fraud in rating, percentage of fraud in review and percentage of fraud in ranking by which the user can decide the application to download

## V. CONCLUSION

Specifically, we first showed that ranking fraud happened in leading periods and provided a process for mining leading periods for each App from its old rating records.

## VI. FUTURE WORK

Here we show how to identify ranking based evidences, rating based evidences and experience based evidences for detecting ranking fraud. Then we are integrating application fraud detection method to make procedure robust.

## REFERENCES

- [1] K. Shi and K. Ali, "Getjar Mobile Application Recommendations with Very Sparse Datasets", International Conference on Knowledge Discovery and Data Mining, 2012.
- [2] N. Spirin and J. Han, "Survey On Web Spam Detection Principles and Algorithms", SIGKDD Explor, 2012.



ISSN(Online) : 2319-8753  
ISSN (Print) : 2347-6710

## International Journal of Innovative Research in Science, Engineering and Technology

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 5, Issue 4, April 2016**

- [3] M. N. Volkovs and R. S. Zemel, "A Flexible Generative Model for Preference Aggregation", International Conference on World Wide Web, 2012.
- [4] Clifton Phua, Vincent Lee, Kate Smith and Ross Gayler, "A Comprehensive Survey of Data Mining-based Fraud Detection Research".
- [5] Z.Wu, J.Wu, J. Cao, and D. Tao Hysad, "A Semi-Supervised Hybrid Shilling Attack Detector for Trustworthy Product Recommendation", International Conference on Knowledge Discovery and Data Mining, 2012.
- [6] S. Xie, G. Wang, S. Lin, and P. S. Yu, "Review Spam Detection via Temporal Pattern Discovery", international conference on Knowledge discovery and data mining, 2012.
- [7] B. Yan and G. Chen, "Appjoy: Personalized Mobile Application Discovery", International Conference on Mobile Systems, Applications, and Services, MobiSys, 2011.
- [8] L. Azzopardi, M. Girolami, and K. V. Risjbergen, "Investigating the relationship between language model perplexity and in precision recall measures", In Proceedings of the 26th International Conference on Research and Development in Information Retrieval (SIGIR'03), pp 369–370,2003