ISSN NO. 2320-5407

Journal homepage:http://www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH

RESEARCH ARTICLE

MINING FRAUD DETECTION IN MOBILE APPS BY ANALYZING USER RATINGS AND RANKING.

Mr. C.KarthikME(CSE)[1], Ms. S. Ramya ME(CSE)[2].

1. Department of CSE, PPG Institute of Technology, Coimbatore,Tamilnadu.
2. Department of CSE, PPG Institute of Technology, Coimbatore, Tamilnadu.

Manuscript Info

Manuscript History:

Received: 14 January 2016
Final Accepted: 25 February 2016
Published Online: April 2016

Key words:
Mobile Apps, rating and review, ranking fraud detection, historical ranking records, data aggregation,

Corresponding Author

Mr. C.Karthik.

Abstract

Discovering fraudulent mobile apps in mobile app market method we are going to identify the fraudulent mobile apps by collecting three type of evidence and eliminating the fraud apps. Using this method app user can save their data and time because we are eliminating the mobile apps in our market by collecting rating based evidence, review based evidence and ranking based evidence from user who already download and used. After identifying the fraudulent apps.we are going to eliminate the fraudulent apps. The data aggregation is based on Leading event and Leading session.

Introduction:-

Now-a-days the number of apps are developing by apps developers. Apps developers are developing many kinds of apps for many purpose. In apps store two kind of categories are there. That are “Top paid apps” and “Top free apps”. Now-a-days many fraudulent apps are spreading in apple or apps store. In our project,We are going to find out the fraudulent apps based on three type of evidence, which are rating, review, ranking. And also based on this fraudulent apps are eliminated from our web page and finding out the active user.

Initiallycreating the web page to get apps details by registering apps by developer. Based on the users response admin will collect the evidence using OLAP method we are going to aggregate the data which is collecting from user. After aggregation administrator going to picking out the fraudulent apps and eliminating that apps. Draw the chart using the aggregating data for easy identification. Using this user can identify the fraudulent app and original app. And they can save their data and money which they are paying in case of paid app.

INPUT

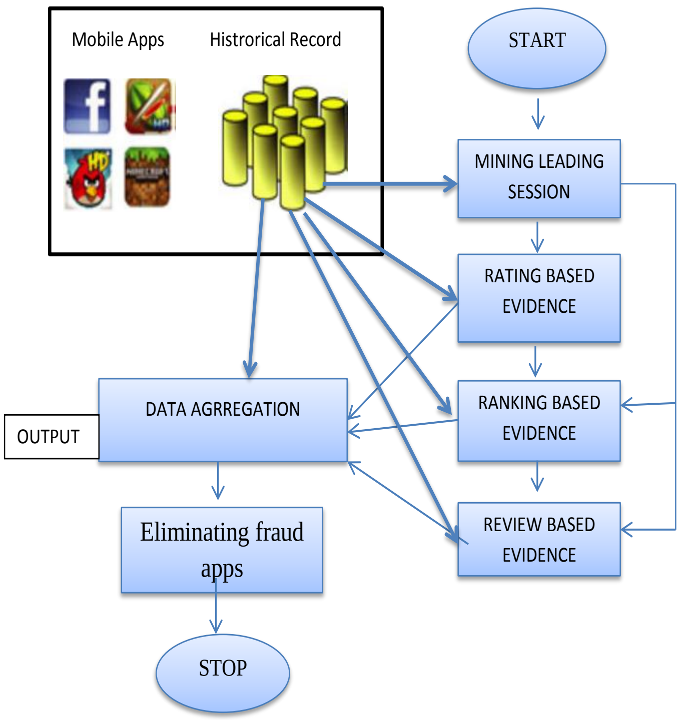

Fig1.1 framework of mining fraud detection in mobile apps

In the literature survey, there are some works related to detection like spam detection, online review spam detection. LatentDirichletallocation describes explanation of challenges of modeling text collection and other collections of discrete data. By using novelty detection, summarization, and similarity and relevance judgments, the large number discrete data are briefly described.In next literature model building is a major task of opinion spam analysis.In Supervised Rank Aggregation, better one will generate by combining results of entity's ranking from multiple ranking functions. Based on these existing we going to proposed that detecting the fraudulent apps in mobile apps market. But we have some challenges in it. Which are Ranking fraud, Rating fraud does not happen always. So we want to detect the fraud when it happens. Detecting Ranking and Rating fraud mobile apps is difficult. So automatic fraud app detection is only way to detect fraud app from multiple Ranking and Rating.

Mining leading event and leading session for mobile apps:-

In this section We are going to identifythe leading event and leading session of mobile apps from their historical Ranking records. We are consider the app leaderboard with respect to "Top paid apps" and "Top free apps". Leaderboard apps of will update periodically for example: Daily bases, weakly bases or constant time period bases. Therefore, each mobileApp has many historical ranking records, mobile App $a$can bedenoted as a time series, $R_a=\{r_1^a, \dots r_i^a, \dots r_n^a\}$, Where $r_i^a \in \{1, \dots, K, \infty\}$ is the ranking of app $a$at time stampt_i. $+\infty$denotes app $a$is not ranked in the top list (K).n denotes the numberof all ranking records. Note that r_i^a is a smaller value. It has higher ranking position.By analyzing the historical ranking records of mobileAppsare not always highly ranked in the leaderboard but only in some leading events.

Definition 1 (leading Event):-

Ranking threshold $K^*\in[1,K]$,Leading event (e) of a contain time limit $T_e=[t^e_{start}, t^e_{end}]$ and ranking of a.which satisfies $r^a_{start} \le K^* < r^a_{start-1}$,and$r^a_{end} \le K^* < r^a_{end+1}$, Moreover ,$\forall t_k \in (t^e_{start}, t^e_{end})$, we have $r_k^a \le K^*$.

we apply apps a ranking thresholdK* which is usuallysmaller than K here K may be very big (e.g., morethan 2,000), and records below K*(e.g., 100).which are not very useful for detecting the ranking manipulations.

Definition 1 (leading Session):-
A leading session s of app a time range $T_s=[t^s_{start}, t^s_{end}]$ and nadjacent leadingEvents $\{e_1,\ldots,e_n\}$, which satisfies $t^s_{start} = t^{e1}_{start}$, $t^s_{end} = t^{en}_{start}$.and there is no other leading session s* that makes $T_s \leq T_{s*}$.meanwhile,$\forall_i \in [1.n)$, we have $(t^{e+1}_{start} - t^e_{end}) < \emptyset$.

Mining leading session:-
In this section we are mining the session valuesleading event€ and session (s) as tuples <∈(t^e_{start} ,t^e_{end}> and <t^s_{start} ,t^s_{end}>respectively, where E_s is the set of leading events insession s.

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: $S_a = \emptyset$;

$E_s = \emptyset$; $e = \emptyset$; $s = \emptyset$; $t^e_{start} = 0$;
for each $i \in [1, |R_a|]$ **do**
  if $r^a_i \leq K^*$ **and** $t^e_{start} == 0$ **then**
    $t^e_{start} = t_i$;
  else if $r^a_i > K^*$ **and** $t^e_{start} \neq 0$ **then**
    //found one event;
    $t^e_{end} = t_{i-1}$; $e =< t^e_{start}, t^e_{end} >$;
    if $E_s == \emptyset$ **then**
      $E_s \cup = e$; $t^s_{start} = t^e_{start}$; $t^s_{end} = t^e_{end}$;
    else if $(t^e_{start} - t^s_{end}) < \phi$ **then**
      $E_s \cup = e$; $t^s_{end} = t^e_{end}$;
    else then
      //found one session;
      $s =< t^s_{start}, t^s_{end}, E_s >$;
      $S_a \cup = s$; $s = \emptyset$ is a new session;
      $E_s = \{e\}$; $t^s_{start} = t^e_{start}$; $t^s_{end} = t^e_{end}$;
    $t^e_{start} = 0$; $e = \emptyset$ is a new leading event;
return S_a

Data aggregation:-
By using WRP Algorithm data aggregation in perform, Based on the page popularity by taking into consideration of both the outlinks and inlinks of the pages WPR decides the rank score.WRP algorithm provides high value of rank and doesn't equally divide the rank of a page among it'soutlink pages. Based on its popularity only every outlink page giving a rank value.First step of the Algorithm is Find rich hyperlinks website, because the weighted PageRank and WPR (VOL) methods rely on the web structures. Next Then generate the web map from the selected website. Then Win (v,u) for each node present in web graph is calculated by applying the equation.

$$W^{in}_{(m,n)} = \frac{I_n}{\sum_{p \in R(m)} I_p}$$

Where
Based on the number of inlinks of all reference pages of page v,Win(v,u) is the weight of link(v, u) calculated. the number of inlinks of page u and.

R (m) denotes the reference page list of page m.

By using the proposed formula to calculate the PageRank value of the nodes present in web graph

$$WPR_{vol}(u) = (1-d) + d\sum_{v \in B(u)} \frac{L_u WPR_{vol}(v) W^{in}_{(v,u)}}{TL(v)}$$

Where

- u represents a web page,
- B(u) is the set of pages that point to u,
- d, is the dampening factor.
- WPR_{vol} (u) and WPR_{vol} (v) are rank scores of page u and v respectively,
- L_u denotes number of visits of link which is pointing page u form v.
- TL(v) denotes total number of visits of all links present on v.

Repeat proposed formula final step will be used recursively until the values are to be stable.

Reference:-

1. (2014).[Online].Available:http://en.wikipedia.org/wiki/information_retrieval.
2. L. Azzopardi, 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.
3. D. M. Blei, and M. I. Jordan, “Latent Dirichlet allocation,” J. Mach. Learn. Res., pp. 993–1022, 2003.
4. Y. Ge, H. Xiongand Z.-H. Zhou, “A taxi driving fraud detection system,” in Proc. IEEE 11th Int. Conf. Data Mining, 2011, pp. 181–190
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. Klementiev,D. Roth, and K. Small, “Unsupervised rank aggregation with distance-based models,” in Proc. 25th Int. Conf. Mach. Learn., 2008, pp. 472–479.
8. 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.
9. 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.
10. 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.