**International Journal of Technical Research and Applications** e-ISSN: 2320-8163,
www.ijtra.com, Volume 5, Issue 3 (May-June, 2017), PP. 22-30

# DISCOVERY OF FRAUD RANKING FOR MOBILE APPS

**[1] Chate Shrikrishna S., [2] Prof. V. R. Chirchi**

[1] ME(CSIT),2nd Year, MBES, COE, Ambejogai-431517, Maharashtra, India
[2] Assistant Professor, Dept. of CSE, MBES, COE, Ambejogai-431517, India.
[1] krishna.chate@gmail.com, [2] vr.chirchi@gmail.com

***Abstract* — 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 develops 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 investigate two types of evidences, ranking based evidences and rating based evidences, by modeling Apps' ranking and rating 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 Apple's 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.**

**General Terms**
**Database Applications - Data Mining.**

***Index Terms* — Ranking Fraud Detection, Mobile Apps.**

## 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 the 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 advertising campaigns 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, 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 "bot farms" or "human water armies" to inflate the App downloads and ratings in a very short time. For example, an article from VentureBeat [2] 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 50,000-100,000 new 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 [2] in the Apple's App store.

In the literature, while there are some related work, such as web ranking spam detection [10, 12, 13], online review spam detection [9, 14, 15], and mobile App recommendation [11, 16, 17, 18], 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. 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 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.

Indeed, our careful observation reveals that fraudulent Apps do not always be 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. Specifically, we first propose a simple yet effective algorithm to identify the leading sessions of each App based on its historical ranking records. Then, with the analysis of Apps' ranking behaviors, we find that the fraudulent Apps often have different ranking patterns in each leading session com-pared 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. Nonetheless, the ranking based evidences can be affected by 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 pro-pose two functions to discover rating based evidences, which reflect some anomaly patterns from Apps' historical rating records. In addition, we develop an unsupervised evidence-aggregation method to integrate these two types of evidences for evaluating the credibility of leading sessions from mobile Apps. Figure 1 shows the framework of our ranking fraud detection system for mobile Apps.

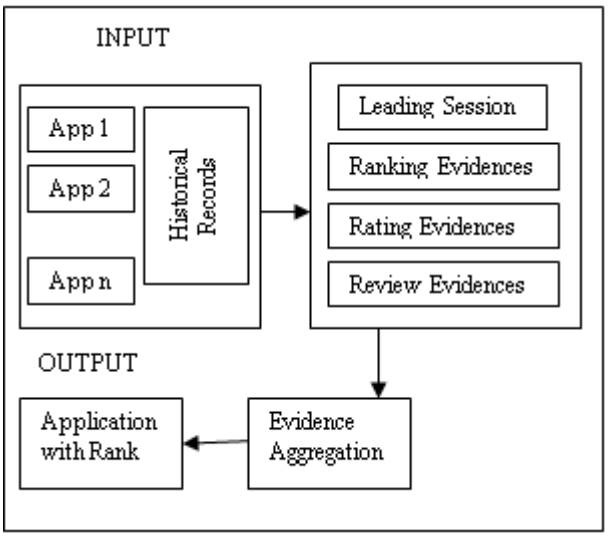


Figure 1: System Framework

The proposed framework is scalable and can be extended with other domain-generated evidences for ranking fraud detection. Finally, we evaluate the pro-posed system with real-world App data collected from the Apple's App store for a long time period. Experimental results show the effectiveness of the proposed system, the scalability of the detection algorithm as well as some regularity of ranking fraud activities.

A. Existing System

In the literature, while there are some related work, such as web ranking spam detection, online review spam detection and mobile App recommendation, the problem of detecting ranking fraud for mobile Apps is still under-explored.

Generally speaking, the related works of this study can be grouped into three categories.

The first category is about web ranking spam detection.

The second category is focused on detecting online review spam.

Finally, the third category includes the studies on mobile App recommendation.

B. Proposed System

We first propose a simple yet effective algorithm to identify the leading sessions of each App based on its historical ranking records. 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.

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 Ranking Based Evidences, by analyzing the Apps' historical ranking records, we observe that Apps' ranking behaviors in a leading event always satisfy a specific ranking pattern, which consists of three different ranking phases, namely, rising phase, maintaining phase and recession phase.

In Rating Based Evidences, specifically, after an App has been published, it can be rated by any user who downloaded it. Indeed, user rating is one of the most important features of App advertisement. An App which has higher rating may attract more users to download and can also be ranked higher in the leaderboard. Thus, rating manipulation is also an important perspective of ranking fraud.

In Review Based Evidences, besides ratings, most of the App stores also allow users to write some textual comments as App reviews. Such reviews can reflect the personal perceptions and usage experiences of existing users for particular mobile Apps. Indeed, review manipulation is one of the most important perspectives of App ranking fraud.

Overview. The remainder of this paper is organized as follows. In Section 2, we introduce some preliminaries and how to mine leading sessions for mobile Apps. Section 3 presents how to extract ranking and rating based evidences and combine them for ranking fraud detection. In Section 4 mathematical model is shown about the proposed approach. In Section 5, we report the experimental results on long-term data sets. Section 6 provides a brief review of related works. Finally, in Section 7, we conclude the paper.

## II. IDENTIFICATION OF LEADING SESSIONS FOR MOBILE APPS

In this section, we first introduce some preliminaries, and then show how to mine leading sessions for mobile Apps from their historical ranking records.

A. Preliminaries:

The App leaderboard demonstrates top K popular Apps. Moreover, the leaderboard is usually updated periodically (e.g., after every 3 days). Therefore, each mobile App a has many historical ranking records which can be denoted as a time series, Ra = {r1a, r2a ,…rna }, where riaε{1,….,K,+∞} is the ranking of a at time stamp ti; +∞ means a is not ranked in the top K list; n denotes the number of all ranking records. Note that, the smaller value ria has, the higher ranking position the App obtains.

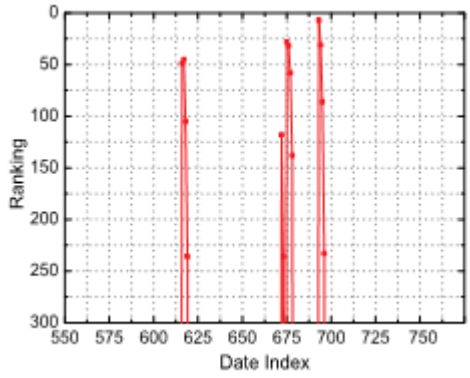

Figure 2a Leading Events

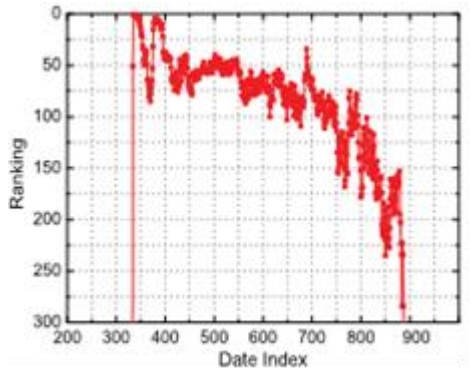

Figure 2b Leading Session

By analyzing the historical ranking records of mobile Apps, we observe that Apps are not always ranked high in the leaderboard, but only in some leading events. For example, Fig. 2a shows an example of leading events of a mobile App. Formally; we define a leading event as follows:

Definition 1 (Leading Event).Given a ranking threshold $K^*\varepsilon[1, K]$, a leading event *e* of App *a* contains a time range $T_e=[t^e_{start}, t^e_{end}]$ and corresponding rankings of *a*, which satisfies $r^a_{start}\leq K^* < r^a_{start-1}$, and $r^a_{end}\leq K^* < r^a_{end+1}$. Moreover, $\forall t_k\varepsilon( t^e_{start}, t^e_{end})$, we have $r_k{}^a\leq K^*$.
Note that we apply a ranking threshold $K^*$ which is usually smaller than K here because K may be very big, and the ranking records beyond $K^*$ 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. For example, Fig. 2b shows an example of adjacent leading events of a given mobile App, which form two leading sessions. Particularly, a leading event which does not have other nearby neighbors can also be treated as a special leading session. The formal definition of leading session is as follows:

Definition 2 (Leading Session).A leading session *s* of App a contains a time range $T_s=[t^s_{start}, t^s_{end}]$ and *n* adjacent leading events $\{e_1,\ldots,e_n\}$, which satisfies $t^s_{start}= t^{e1}{}_{start}$, $t^s_{end}= t^{en}{}_{end}$ and there is no other leading session $s^*$ that makes $T_s\leq T_{s*}$. Meanwhile, $\forall i\ \varepsilon[1,n)$, we have $(t^{ei+1}_{start}\text{-}t^{ei}_{end})<\emptyset$, where $\emptyset$ is a predefined time threshold for merging leading events.
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.

B. Mining Leading Sessions

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 1 demonstrates the pseudo code of mining leading sessions for a given App ***a***.

**Algorithm 1** pulling out Leading Sessions
**Input 1: a**'s historical ranking records $R_a$;
**Input 2:** the ranking threshold $K^*$;
**Input 3:** the merging threshold $\emptyset$;
**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** $r_i^a \leq K^*$ **and** $t^e_{start}==0$ **then**
4: $t^e_{start} = t_i$;
5: **else if** $c > K^*$ **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: $E_s \cup= e$; $t^e_{start} = t^e_{start}$; $t^e_{end} = t^e_{end}$;
10: **else if** $(t^e_{start} - t^e_{end}) < \emptyset$ **then**
11: $E_s \cup= e$; $t^e_{end} = t^e_{end}$;
12:**else then**
13: // found one session:
14: $s$ = < $t^e_{start}$, $t^e_{end}$, $E_s$ >;
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 < $t^e_{start}$, $t^e_{end}$ > and < $t^s_{start}$, $t^s_{end}$ ,$E_s$> respectively, where $E_s$ 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 $(t^e_{start}\text{-}t^s_{end}) < \emptyset$, *e* will be considered as a new leading session (i.e., Step 8 to16). Thus, this algorithm can

identify leading events and sessions by scanning ***a***'s historical ranking records only once.

## III. MODULES

- Ranking Based Evidences
- Rating Based Evidences
- Review Based Evidences
- Evidence Aggregation

### A. Ranking Based Evidences

In this module, we develop Ranking based Evidences system. By analyzing the Apps' historical ranking records, web serve that Apps' ranking behaviors in a leading event always satisfy a specific ranking pattern, which consists of three different ranking phases, namely, rising phase, maintaining phase and recession phase. Specifically, in each leading event, an App's ranking first increases to a peak position in the leaderboard (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 (i.e., recession phase). Fig.4.1 shows the different ranking phases of a leading event.

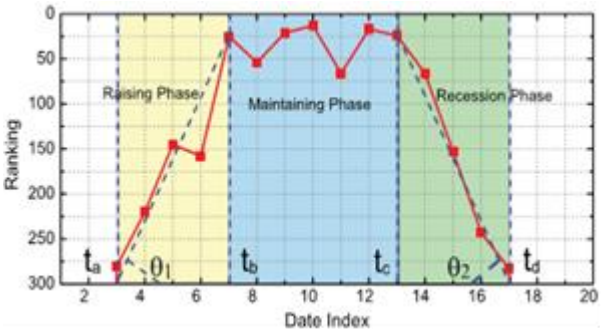


Figure.3 Different Ranking Phases

In ranking based evidences specific ranking pattern is always satisfied by app ranking behavior. This includes rising phase, maintaining phase and recession phase.

*Evidence 1:*

Ranking pattern for rising and recession phases:-

$$\theta_1^e = \arctan\left(\frac{K^* - r_b^a}{t_b^e - t_a^e}\right), \theta_2^e = \arctan\left(\frac{K^* - r_c^a}{t_d^e - t_c^e}\right) \quad (1)$$

Fraud signature for leading session:-

$$\bar{\theta}_s = \frac{1}{|E_s|} \sum_{e \in s} \theta_1^e + \theta_2^e \quad (2)$$

*Evidence 2:*

Ranking pattern for maintaining phase:-

$$\Delta t_m^e = (t_c^e - t_b^e + 1) \quad (3)$$

Fraud signature for leading session:-

$$X_s = \frac{1}{|E_s|} \sum_{e \in s} \frac{K^* - \bar{r}_m^e}{\Delta t_m^e} \quad (4)$$

### B. Rating Based Evidences

In the third module, we enhance the system with Rating based evidences module. The ranking based evidences are useful for ranking fraud detection. However, sometimes, it is not sufficient to only use ranking based evidences. For example, some Apps created by the famous developers, such as Gameloft, may have some leading events with large values of u1 due to the developers' credibility and the "word-of-mouth" advertising effect. Moreover, some of the legal marketing services, such as "limited-time discount", may also result in significant ranking based evidences. To solve this issue, we also study how to extract fraud evidences from Apps' historical rating records. In rating based evidences rating pattern is used for ranking fraud detection in app. This rating is done after downloading the app by user and then user gives rating to that app. If the rating is high in the leader board of app industry then that app is attracted by more mobile app users. In this the fraud occurred during rating is performed in leading session. An App with rating fraud might have surprisingly high ratings in the fraudulent leading sessions.

Evidence 3:Fraud signature: $\Delta R_s = \frac{\bar{R}_s - \bar{R}_a}{\bar{R}_a}$, $(s \in a)$ (5)

### C. Review Based Evidences

In this module we add the Review based Evidences module in our system. Besides ratings, most of the App stores also allow users to write some textual comments as App reviews. Such reviews can reflect the personal perceptions and usage experiences of existing users for particular mobile Apps. Indeed, review manipulation is one of the most important perspectives of App ranking fraud. Specifically, before downloading or purchasing a new mobile App, users often first read its historical reviews to ease their decision making, and a mobile App contains more positive reviews may attract more users to download. Therefore, imposters often posts fake reviews in the leading sessions of a specific App in order to inflate the App downloads and thus propel the App's ranking position in the leader board.

### D. Evidence Aggregation

In this module we develop the Evidence Aggregation module to our system. After extracting three types of fraud evidences, the next challenge is how to combine them for ranking fraud detection. Indeed, there are many ranking and evidence aggregation methods in the literature, such as permutation based models score based models and Dempster-Shafer rules. However, some of these methods focus on learning a global ranking for all candidates. This is not proper for detecting ranking fraud for new Apps. Other methods are based on supervised learning techniques, which depend on the labeled training data and are hard to be exploited. Instead, we propose an unsupervised approach based on fraud similarity to combine these evidences.

## IV. MATHEMATICAL MODEL

Let S, be a system that describes detection of ranking Fraud for Mobile Apps- S= {I, P, O}
Where,

**1) Input (I):** Historical data for Apps,
I = {$i_1$, $i_2$, $i_3$, $i_4$, $i_5$}; where,
$i_1$ = App Name, $i_2$ = Upload, $i_3$ = Download, $i_4$ = Rating, $i_5$ = Review

**2) Process (P)** = {$p_1$, $p_2$, $p_3$, $p_4$, $p_5$}; where
$p_1$ = MLS;

$R_a = \{r_1^a, r_2^a, \ldots, r_n^a\}$;

$r_i^a = \{1, K, \ldots, +\infty\}$; where

$R_a$ = is a's historical ranking records,

$r_i^a$ = is ranking of a at time $t_i$;

$+\infty$ = App is not ranked in top K;

n = number of all ranking records;

$p_2 = R_nBE$
I] For Rising and Recession Phase;

$\overline{\theta_s} = \frac{1}{|E_s|}\sum_{e \in s}(\theta_1^e + \theta_2^e)$, where,

$\overline{\theta}_s$ = fraud signature of s;

$\theta_1^e, \theta_2^e$ = shape parameter from Eq. (1) & Eq. (2);

$|E_s|$ = number of e's in session s

II] For Maintaining Phase;

$X_s = \frac{1}{|E_s|}\sum_{e \in s}\frac{K^* - \bar{r}_m^e}{\Delta t_m^e}$, where,

$X_s$ = fraud signature for s;
$K^*$ = ranking threshold;
$\bar{r}_m^e$ = average rank in this phase;
$\Delta t_m^e$ = maintaining phase from Eq. (3)

$p_3 = R_tBE$

$\Delta R_s = \frac{\bar{R}_s - \bar{R}_a}{\bar{R}_a}$, $(s \in a)$, where

$\Delta R_s$ = fraud signature;

$\bar{R}_s$ = average rating in leading session s;

$\bar{R}_a$ = avg. rating of app a

$p_4 = R_eBE$
Reviews analysis for review based evidences

$p_5$ = EA
Linear combination of all existing evidences

**3) Output(O):** {$o_1$, $o_2$, $o_3$, $o_4$}, where
$o_1$ = Top K-ranked apps;
$o_2$ = Historical ranking;
$o_3$ = Evidence details;
$o_4$ = App review

## V. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of ranking fraud detection using Apps data.

### A. Experimental Data

The experimental data is collected from our system. The data set contain the periodical chart ranking of Apps. Moreover dataset also contains the user rating and review information.

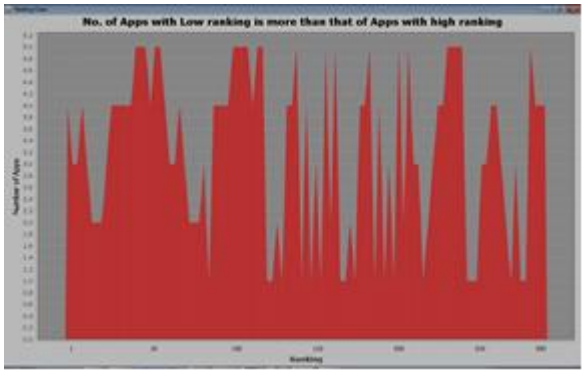


Figure.4 distribution of the number of Apps w.r.t. different rankings

Fig. 4 shows the distributions of the number of Apps with respect to different rankings in these data sets. In the figures, we can see that the number of Apps with low rankings is more than that of Apps with high rankings.

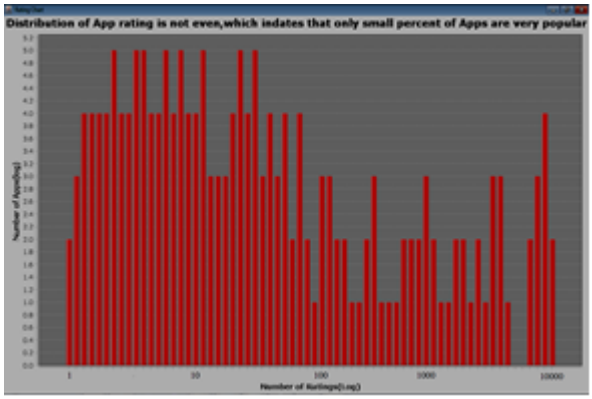


Figure.5 The distribution of number of Apps w.r.t. different numbers of ratings

Fig.5 shows the distribution of the number of Apps with respect to different number of ratings in the data set. In the figures, we can see that the distribution of App ratings is not even, which indicates that only a small percentage of Apps are very popular.

### B. Mining Leading Sessions

Here, we demonstrate the results of mining leading sessions in data set. Specifically, in Algorithm 1, we set the ranking threshold K* = 92 and threshold = 6. This denotes two adjacent leading events can be segmented into the same leading session if they occur within one week of each other.

Fig 6 and Fig 7 show the distributions of the number of Apps with respect to different numbers of contained leading events and leading sessions in data set. In these figures, we can see that only a few Apps have many leading events and leading sessions. The average numbers of leading events and leading sessions are 5.1 and 2.62 for the Apps

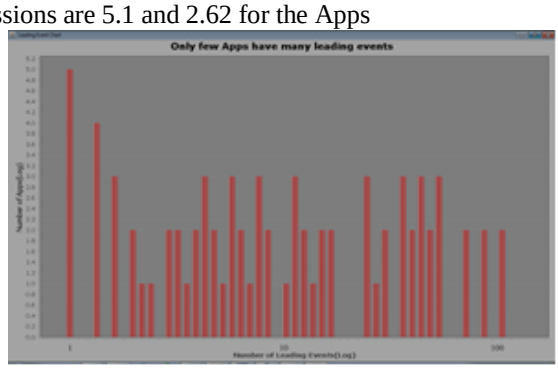


Figure.6 Distribution no. of Apps w.r.t different no. of leading events

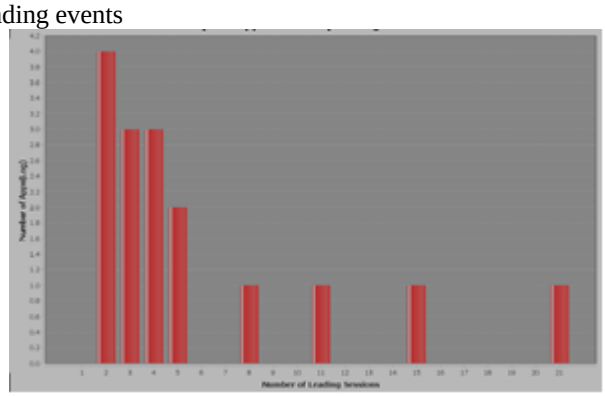

Figure.7 Distribution no. of Apps w.r.t different no. of leading sessions

Fig. 8 shows the distribution of the number of leading sessions with respect to different numbers of contained leading events in both data sets. In these figures, we can find only a few leading sessions contain many leading events. Indeed, the average number of leading events in each leading session is 2.26for the Apps.

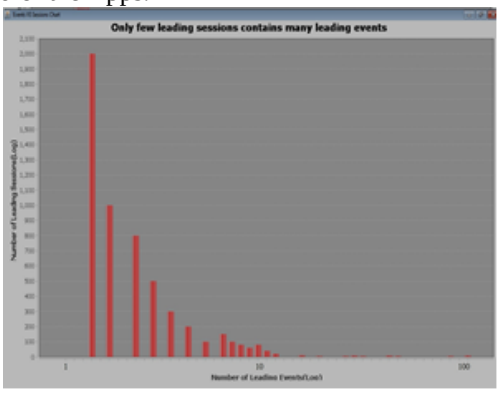


Figure.8 The distribution of the no. of leading sessions w.r.t different no. of leading events.


Figure.9 Download rating for our System Apps

C. Evaluation on Human Judgment

To the best of our knowledge, there is no existing bench-mark to decide which leading sessions or Apps really contain ranking fraud. Thus, we develop three intuitive base-lines and invite five human evaluators to validate the effectiveness of our approach EA-RFD (Evidence Aggregation Based Ranking Fraud Detection).

1) Baselines

The first baseline Ranking-RFD stands for Ranking evidence based Ranking Fraud Detection, which estimates ranking fraud for each leading session by only using ranking based evidences.

The second baseline Rating-RFD stands for Rating evidence based Ranking Fraud Detection, which estimates the ranking fraud for each leading session by only using rating based evidences

Above two baselines are used for evaluating the effectiveness of different kinds of evidences.

Note that, we need to define some ranking ranges before extracting ranking based evidences for EA-RFD and Rank-RFD. In our experiments, we segment the rankings into 5 different ranges, i.e., [1, 10], [11, 25], [26, 50], [51, 100], [101, 300], which are commonly used in App leaderboards.

2) Experimental Setup

To study the performance of ranking fraud detection by each approach, we set up the evaluation as follows.

First, for each approach, we selected 20 top ranked leading sessions (i.e., most suspicious sessions), and 20 bottom ranked leading sessions (i.e., most normal sessions) from data set. Then, we merged all the selected sessions into a pool which consists 114 unique sessions from 84 unique Apps in data set.

Second, we invited two human evaluators who are familiar with Apple's App store and mobile Apps to manually label the selected leading sessions with score 1 (i.e., Fraud) and 0 (i.e., Non-fraud). Specifically, for each selected leading session, each evaluator gave a proper score by comprehensively considering the profile information of the App (e.g., descriptions, screenshots), the trend of rankings during this

session, the App leaderboard information during this session, the trend of ratings during this session, and the user comments during this session. Moreover, they can also download and try the corresponding Apps for obtaining user experiences. Particularly, to facilitate their evaluation, we develop a Ranking Fraud Detection System, which ensures that the evaluators can easily browse all the information. Also, the platform demonstrates each leading session in random orders, which guarantees there is no relationship between leading sessions' order and their fraud scores.

Third, after human evaluation, each leading sessions is assigned a fraud score f(s) ∈ [0, 5]. Finally, we further ranked the leading sessions by each approach with respect to their fraudulent scores, and obtained six ranked lists of leading sessions. In particular, if we treat the commonly agreed fraud sessions (i.e.,89 sessions in data set) as the ground truth, we can evaluate each approach with three widely-used metrics, namely Precision@K, Recall@K, F@K [2].

Also, we can exploit the metric normalized discounted cumulative gain (NDCG) for determining the ranking performance of each approach. Specifically, the discounted cumulative gain given a cut-off rank K can be calculated by

$$DCG@K = \sum_{i=1}^{K} \frac{2^{f(si)}-1}{log2(1+i)},$$

Where, $f(s_i)$ is the human labeled fraud score. The NDCG@K is the DCG@K normalized by the IDCG@K, which the DCG@K value of the ideal is ranking list of the returned results, i.e., we have $NDCG@K = \frac{DCG@K}{IDCG@K}$,NDCG@K indicates how well the rank order of given sessions returned by an approach with a cut-off rank K. The large value of NDCG@K, the better performance of ranking fraud detection.

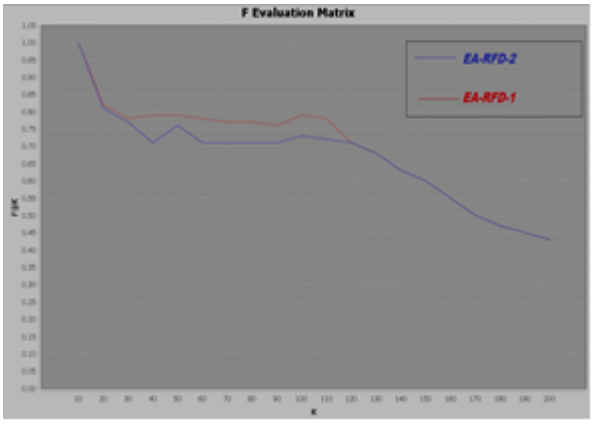


Figure 10(a) F@K results of each approach

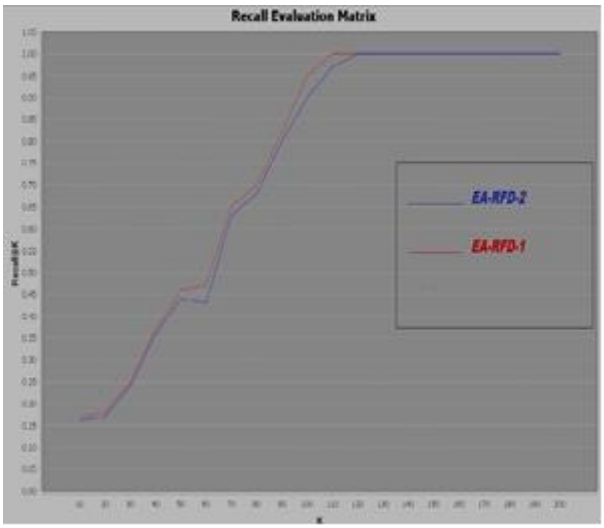


Figure 10(b) The Recall@K results of each approach

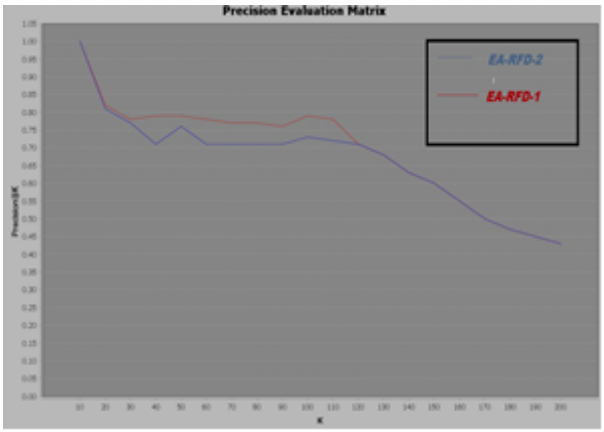


Figure 10(c) Precision@K results of each approach

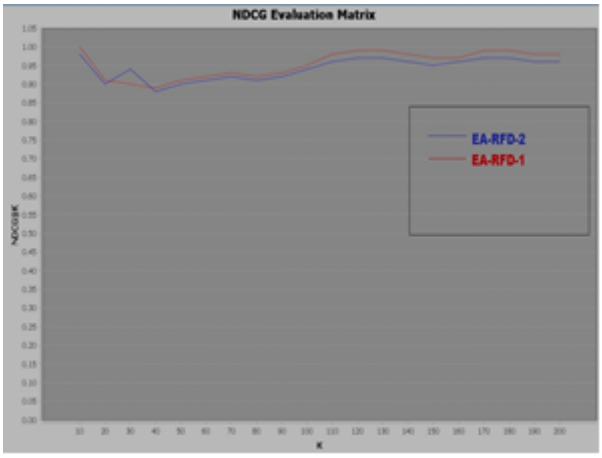


Figure 10(d) NDCG@K results of each approach

3) Overall Performance

In this section, we present the overall performances of each ranking fraud detection approach with respect to different evaluation metrics, i.e., Precision@ K , Recall@ K , F@k, and NDCG@K . Particularly, here we set the maximum K to be 200, and all experiments are conducted on a 2.8 GHZ*2 quad-core CPU, 4G main memory PC.

Fig.10. show the evaluation performance of each detection approach in data set. From these figures we can observe that the evaluation results in data sets are consistent. Indeed, by analyzing the evaluation results, we can obtain several insightful observations. Specifically, first, we find that our approach, i.e., EA-RFD-2/EA-RFD-1, consistently outperforms other baselines and the improvements are more significant for smaller K (e.g., K<100). This result clearly validates the effectiveness of our evidence aggregation based framework for detecting ranking fraud. Second, EA-RFD-2 outperforms EA-RFD-1 slightly in terms of all evaluation metrics, which indicates that rank based aggregation (i.e., Principle 2) is more effective than score based aggregation (i.e., Principle 1) for integrating fraud evidences. This indicates that leveraging kind of evidences are more effective than only using one type of evidences, even if without evidence aggregation. Finally, by comparing Ranking-RFD and Rating-RFD, we can observe that the ranking based evidences are more effective than rating and review based evidences. It is because rating and review manipulations are only supplementary to ranking manipulation.
To further validate the experimental results, we also con-duct a series of paired T-test of 0.95 confidence level which show that the improvements of our approach, i.e., EA-RFD-2/EA-RFD-1, on all evaluation metrics with different K compared to other baselines are all statistically significant.

D. Case Study: Evaluation of Apps Credibility

Here, we study the performance of evaluating App credibility based on the prior knowledge from existing reports. Specifically, as reported by IBTimes [4], there are eight free Apps which might involve in ranking fraud. In this paper, we use seven of them in our data set (Tiny Pets, Social Girl, Fluff Friends, Crime City, VIP Poker, Sweet Shop, Top Girl) for evaluation. Indeed, we try to study whether each approach can find these suspicious Apps with high rankings, since a good ranking fraud detection system should have the capability of capturing these suspicious Apps.

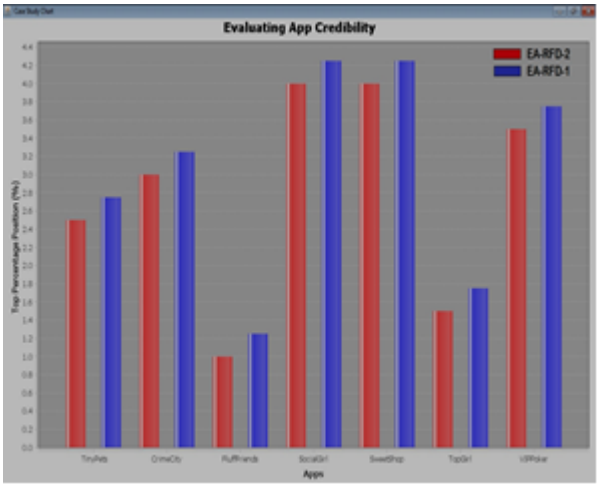


Figure 11 Case study of reported suspicious mobile Apps

Fig. 14 shows the top percentage position of each App in the ranked list returned by each approach. We can see that our approach, i.e., EA-RFD-2 and EA-RFD-1, can rank those suspicious Apps into higher positions than other baseline methods. Similarly as the results in Section 5.3.3, only leveraging single kind of evidences for fraud detection can-not obtain good performance, i.e., finding such suspicious Apps in high positions.

## VI. RELATED WORK

This paper aims to detect users generating spam reviews or review spammers. They identify several characteristic behaviors of review spammers and model these behaviors so as to detect the spammers. 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. Their results show that our proposed ranking and supervised methods are effective in discovering spammer sand outperform other baseline method based on helpfulness votes alone. They 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. [13]

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. In this paper, they develop a taxi driving fraud detection system, which is able to systematically investigate taxi driving fraud. In this system, they 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. Finally, they evaluate the taxi driving fraud detection system with large scale real-world taxi GPS logs. In the experiments, they 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 [8]

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. In this paper, they 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. 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. [11]

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. [12]

## 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 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 Apple's App store. Experimental results showed the effectiveness of the proposed approach. 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.

## 8. ACKNOWLEDGMENT

We are glad to express our sentiments of gratitude to all who rendered their valuable guidance to us. We would like to express our appreciation and thanks to the Principal of our college. We are also thankful to the Dean of PG Department. We thank to the anonymous reviewers for their valuable comments.

## REFERENCES

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

[2] (2012). [Online]. Available: https://developer.apple.com/news/index.php?id=02062012a

[3] (2012). [Online]. Available: http://venturebeat.com/2012/07/03/apples-crackdown-on-app-ranking-manipulation/

[4] (2012).[Online].Available: http://www.ibtimes.com/applethreatens- crackdown-biggest-app-store-ranking-fra ud-406764

[5] (2012). [Online]. Available: http://www.lextek.com/manuals/onix/index.html

[6] (2012). [Online]. Available: http://www.ling.gu.se/lager/mogul/porter-stemmer.

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

[8] 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.

[9] 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.

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

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

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

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

[18] Hengshu Zhu, Hui Xiong, Yong Ge, and Enhong Chen, "Discovery of Ranking Fraud for Mobile Apps", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 27, NO. 1, JANUARY 2015

[19] 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.

[20] 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.