ISSN:0975-9646

Security Awareness of Mobile Application for Discovering Fraud Rank

S.K.Ram Kumar[#1], Dr. N. Lakshmi Narasimman[*2]

[#1]*PG Scholar, Computer Science Engineering Department,*
K.L.N. College of Engineering, Pottapalayam, Sivagangai 630 612, INDIA

[*2]*Professor and Head of Dept., Computer Science Engineering,*
K.L.N. College of Engineering, Pottapalayam, Sivagangai 630 612, INDIA

***Abstract*- In this paper, we will discuss the various methods and techniques used for the finding the fraud rankings in various mobile applications. Rankings play a major role in mobile applications and other products. Customers buy applications based on the ranking and also by reading the review and rating given to it. The review and rating are game changers for the success or failure of the application or sales of a product. Mobile Apps are highly varied and often poorly understood, particularly for their activities and functions related to privacy and security. More mobile users are reluctant to adopt mobile Apps due to the risk of privacy invasion and other security concerns. Therefore, we first develop the techniques to automatically detect the potential security risk for each mobile App by exploiting the requested permissions. Specifically, we first investigate evidences such as ranking, review and rating based evidences, by modeling Apps' their behaviours through statistical hypotheses tests. At last, to integrate all the evidences for fraud detection optimization based aggregation method. In this investigation, the paper supports the adequacy of proposed framework and demonstrates the identification of fraud ranking.**

Keywords - Mobile Application, Recommender Systems, Security and Privacy, Ranking, Review and Rating based evidences.

1. INTRODUCTION

Basically the mobile app is a computer program; it is deliberate to scurry on the smart phones, tablet or computer and supplementary mobile phones. The apps are typically pertinent through the application distribution platform, which begin appearing in 2008. For the smart phones there are many mobile apps available. A few of the apps are free of charge, at the same time as others have to be bought.

Usually the apps are downloaded from this platform to a target device, such as iPhone, blackberry, android phone or windows mobile, but sometimes they can be downloaded to a laptop or computer system. A mobile application is most commonly referred to as an app. The 20-30% percentage of app price goes to the distribution provider E.g. iTunes, and the remaining goes to the creator of the app. The mobile apps were initially vacant for the universal production and information retrieval, including calendar, contact, email, stock market and about the weather information. The app availability based on the public demand, so the designer tools herd fast and rapid extension of mobile app into various other kind such as games in mobile, factory computerization, services based on GPS and based on location, banking, ticket purchasing, etc. There is a challenging issue in mobile app recommendation, because of sudden increase in quantity and the variety of mobile apps which in turn led to the conception of broad range of review and creation sources including blogs, magazines and online app services.

To increase the development of mobile Apps, many App stores launched daily App leader boards, which display chart rankings of the highly popular Apps. Certainly, the App leader board is one of the most important ways for promoting the mobile Apps. If an App had got higher rank on the leader board then it leads to more number of downloads which in turn over the revenue of that company into millions of dollars. Therefore, App developers tend to explore many methods such as advertising campaigns to elevate their Apps in order to increase their Apps ranking as high as possible. However shady App developers find some fraudulent means to purposely boost their Apps which eventually manipulate the chart rankings on an App store. This is usually administered by using so-called "human water armies" or App farm which is also called as "bot farm" to increase the App downloads, reviews and ratings in a very short time. For example, app manipulation farm [6] is a place where developers had to pay for their app's download many times so as to artificially inflate the ranking. Another example, an article from Venture Beat [4] reported that, when an App was promoted with this method, it could be propelled from number 1,900 to the top 20 in Apple's top free leader board and more than 50,000 to 100,000 new users could be gained within few days. In fact, such ranking fraud raises great issues for the mobile App industry.

To this end, in this paper, we introduce the method to develop a mobile App recommender system with security and privacy awareness. The design idea is to prepare the recommender system with the ability to detect automatically and evaluate the privacy and security risks of mobile Apps. However, there are two critical challenges for developing recommender App system with security and privacy awareness. Specifically, the first challenge is how to efficiently analyze the security risks of mobile Apps from the large scale mobile App data. The second challenge is how to derive a balance between the popularity of Apps

and the users concerns about security and privacy. Indeed, our careful observation explains that the possible security risks of the Apps are essentially caused by the data access permissions of each App, such as permissions requested for camera features. Therefore, in this paper, we propose to exploit the permissions requested for detecting the potential security risk of each mobile App.

2. PRELIMINARIES

2.1 IDENTIFYING LEADING SESSIONS FOR MOBILE APPS

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

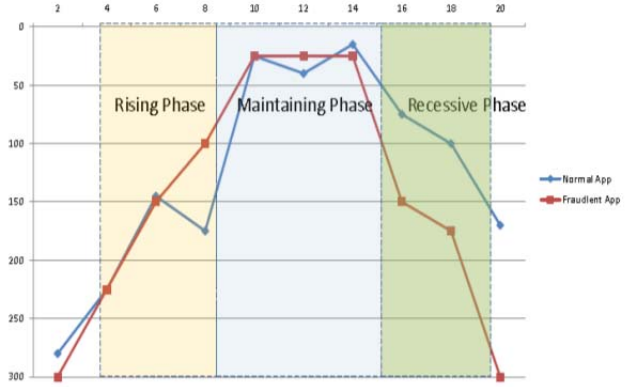

Fig 1. Example of Different ranking phases of a leading event.

A. MINING LEADING EVENTS:

Mining leading events is defined as given a ranking limit K*€ [1,K], leading events 'e' of an mobile App 'a' contains a time period range, Te=[t^e_{start}, t^e_{end}] and followed by rankings of 'a', that satisfies r^a_{start}<K*<r^a_{start}, and r^a_{end}<K*<r^a_{end+1}.

Algorithm 1

Input 1: a's historical ranking records Ra;

Input 2: the ranking threshold K*;

Input 3: the merging threshold Ø;

Output: the set of a's leading sessions S_a;

Initialization: S_a = Ø;

1: Es = Ø; e = Ø; s = Ø; t^e_{start} = 0;

2: for each i ϵ [1, |Ra|] do

3: if $r_i^a \leq K^*$ and t^e_{start} ==0 then

4: $t^e_{start} = t_i$;

5: else if $r_i^a > K^*$ and $t^e_{start} \neq 0$ then

6: $t^e_{end} = ti-1$; e=< t^e_{start}, t^e_{end} >;

7: if Es == Ø then

8: Es U = e; $t^s_{start} = t^e_{start}$; $t^s_{end} = t^e_{end}$;

9: else if ($t^e_{start} - t^s_{end}$) < Ø then

10: Es U = e; $t^s_{end} = t^e_{end}$;

11: else then

12: s = < t^s_{start}, t^s_{end}, Es>;

13: SaU = s; s = Ø is a new session;

14: Es = {e}; $t^s_{start} = t^e_{start}$; $t^s_{end} = t^e_{end}$;

15: t^e_{start} = 0; e = Ø is a new leading event;

16: return S_a

B. MINING LEADING SESSIONS:

There are two main processes for mining leading sessions. First, discover the leading events from the Mobile App's historical ranking records. Second, merge the adjacent leading events for building the leading sessions. Algorithm 1 mining leading sessions describes the pseudo code for the given App 'a'.

2.2 Security/Privacy Problems of Mobile Apps

The Popularity of the App is determined by total number of downloads and average rating. Security for the mobile App is determined based on the access permission.

Type	Permission ID	Description
String	ACCESS_FINE_LOCATION	Allows an application to access fine (e.g., GPS) location.
String	READ_CONTACTS	Allows an application to read the user's contacts data.
String	READ_SMS	Allows an application to read the user's SMS messages.
String	READ_CALENDAR	Allows an application to read the user's calendar data.
String	READ_CALL_LOG	Allows an application to read the user's call log.

Normal permissions:

This gives an App access to confined App level features, with the minimal risk to other applications, the system, or the user access.

Dangerous permissions:

This gives an App access to private data of user or control over the device, with a potential risk that can negatively impact the user.

Signature/System permissions:

This gives an App access to the dangerous level privileges, which need system signature certifications such as the ability to control the overall process of the system.

3. RELATED WORKS

1. Ke Zhai, Jordan Boyd-Graber, "Online Latent Dirichlet Allocation with Infinite Vocabulary"[8]

Jordan Boyd-Graber, Ke Zhai, extends the Latent Dirichlet Allocation by extracting topics from a Dirichlet process whose base distribution is a distribution over all strings rather than from a finite Dirichlet. The Infinite Vocabulary Topic Model is a generative process in which it is identical to LDA's except that topics are not drawn from a finite Dirichlet. Truncation ordered set, where truncating the distribution to a finite subset of all possible atoms. Stochastic inference, the pattern is analyzed statistically and arrives at a conclusion on the basis of evidence. At last refining and updating of the Truncation Ordered Set take place. The result shows that it can add up new words and it performs far better than finite vocabularies topic models in evaluations of classification performance and topic quality.

2. Qi Liu, Hui Xiong, Enhong Chen, Jian Chen, Chris H.Q. Ding, "Enhancing Collaborative Filtering by User Interests Expansion via Personalized Ranking"[1]

Hui Xiong, Enhong Chen, Qi Liu, Jian Chen, Chris H.Q. Ding, exploited user latent interests for developing an item-oriented model-based collaborative framework**.** In iExpand method each user is a probability distribution over interest and each interest is represented as a probability distribution over item. In recommender systems, rating behavior of the user depends on a set of hidden interests. They computed iExpand on user-interest-item data sets, and experimental results show that this performance is better than the state of art methods with a significant margin.

3. H. Peng, C. Gates, B. Sarma, N. Liu, Y. Qi, R. Potharaju, C. Nita-Rotaru, I. Molloy "Using probabilistic generative models for ranking risks of android apps"[3]

Y. Ge, C. Liu, H. Xiong, and Z.-H. Zhou, use probabilistic generative models for risk scoring schemes, and identify several such models, ranging from the simple Naive Bayes, to advanced hierarchical mixture models. Parameter selection is where both MNB and HMNB can be used with different parameters, and selects the best parameters for them to compare with other methods. All generative models have AUC values of over 0.94; they significantly outperform RCP and RPCP. The results clearly show that HMNB is best performing, with MNB, BNB, and PNB close behind and almost the same. Permissions and Risk Scores are calculated.

4. Hengshu Zhu, Huanhuan Cao, Enhong Chen, Hui Xiong, Jilei Tian "Mobile App Classification with Enriched Contextual Information"[2]

Enhong Chen, Huanhuan Cao, Hengshu Zhu, Hui Xiong, Jilei Tian**,** describes the use of mobile Apps which plays a vital role in understanding the user preferences. It provides mobile Apps with the opportunities for enriching the contextual information by discovering the additional Web knowledge from the Web search engine. In Explicit Feedback of Vector Space Model the stop words are removed, build Normalized word vector and calculate the Cosine Similarity. Implicit Feedback of Semantics Topics where the Latent Dirichlet Allocation (LDA) model had been used that allows set of observation that explains about unobserved groups, which explain why few parts of data are similar. It is robust and outperforms two states of the arts baselines.

5. Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, Hady W. Lauw, "Detecting Product Review Spammers using Rating Behaviors"[10]

Viet-An Nguyen, Ee-Peng Lim, Bing Liu, Nitin Jindal, Hady W. Lauw, detect review spammers or users generating spam reviews. Identifying many characteristic behaviors of review spammers and modeling their behaviors for detecting the spammers. In Rating Spamming, the function *sim()* compare the ratings in a given set and it is a similarity function. Reviewers with large proportions of ratings involved as multiple similar ratings on products are to be assigned high spam scores. In Review Text Spamming, user spam's a product with multiple ratings; they are also likely to spam the product with multiple review texts. Define the similarity between two reviews where each review text is represented by a bag of bi-grams and *cosine is the cosine similarity of the bi-gram* TFIDF vectors. TFIDF refers to the frequency *x inverse document frequency of a bi-gram.* In Combined Spam Score, combine the above two spam scores so as to single product having multiple reviews by taking the average. The proposed methods generally outperform the baseline method based on helpfulness votes.

4. SYSTEM ARCHITECTURE:

Mobile Apps are not always highly ranked in the leader board, only in few leading events. This leads to the formation of different leading sessions. In other words, fake ranking usually happens in the leading sessions. Therefore, fake ranking discovery of mobile Apps is actually done to detect fake ranking within leading sessions of mobile Apps.

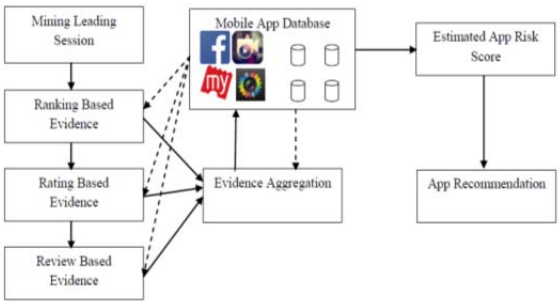

Fig 2. Framework of ranking fraud detection and mobile app Recommender system

The ranking based evidences and App developers' reputation can be affected due to "limited-time discount". Therefore, propose a rating and a review fraud evidences to support the ranking based evidence. Finally all the evidences are integrated by unsupervised evidence-aggregation method.

4.1 Identifying Ranking based evidence:

The Apps ranking behavior in a leading event always follows a specific ranking pattern. It consists of the three phases namely rising, maintaining and recession. For malicious App there will be short rise and short fall.

4.2 Identifying Rating based evidence:

Previous method is useful for detection which is not sufficient. The limited time discount in marketing have a significant effect on ranking based evidence. Rating based evidence is based on time period. Normal App always gets similar average rating each day. If an App has fake ranking in leading session, they will receive much rating in a particular period of time. Compared with the historical ratings, the ratings of the mobile app during the time period of s might have anomaly patterns. These records can be used for obtaining rating based evidences.

4.3 Identifying Review based evidence:

Review which contains some information about the app as comment that had already used plays a vital role in downloading the Apps. A fraud signature is defined based from the previous works. It is computed by removing stop words, build Normalized word vector and calculates the similarity.

4.4 Fraud User

IP address, Session and User Id are used to calculate Fraud User. Each day is separated into 12 sessions.

IP Address	Session	User ID	Fraud / Genuine
Same	Same	Same	Not Possible
Same	**Same**	**Different**	**FRAUD**
Same	Different	Same	Not Possible
Same	Different	Different	GENUINE
Different	Same	Same	Not Possible
Different	Same	Different	GENUINE
Different	Different	Same	Not Possible
Different	Different	Different	GENUINE

If user id is marked fraud once then future rating and review by that user id is not considered. Not possible means a single user Id can give only one input for each app.

4.5 Mobile App Recommendation

To help users understand the different risks of Apps is to categorize the risks into discrete levels (e.g., Low, Medium, and High). In fact, people often describe their perception about risk or security with such discrete levels. Therefore, in The Popularity of the App is determined by total number of downloads and average rating. Intuitively, there are two types of ranking principles for recommending Apps.

RELIABLE	DANGEROUS	SYSTEM
Modify/delete SD card contents	Read Contacts	Make phone calls
Read calendar data	Write contact data	Send SMS or MMS
Write calendar data	Read browser history & bookmarks	Read sensitive logs
Modify global system settings	Write browser history & bookmarks	Authenticate Accounts
Read sync settings	Automatically start at boot	Install DRM
Access mock location	Retrieve running applications	Add system service
Battery stats	Take pictures and videos	In-app billing
Bluetooth Admin	Access location extra commands	Format file systems
Clear app cache	Change Configuration	Process outgoing calls

Security Principle: Ranking of App is evaluated by their risk score in ascending order and the same risk score Apps will be ranked further by popularity scores.

Popularity Principle: Ranking of App is evaluated by their popularity score in descending order and the same popularity score Apps will be ranked further by risk scores.

A. **Estimating Risk Score**

The latent similarity between Apps a_i and a_j by the cosine distance,

$$s^a_{ij} = \mathrm{Cos}(\bar{a}_i, \bar{a}_j) = \frac{\bar{a}_i . \bar{a}_j}{||\bar{a}_i||.||\bar{a}_j||} \quad (1)$$

The latent similarity between Permissions p_i and p_j by the cosine distance,

$$s^p_{ij} = \mathrm{Cos}(p_i, p_j) = \frac{pi.pj}{||p_i||.||p_i||} \quad (2)$$

$$Q(a,p) = \lambda/2 .\{\Sigma_i |R_i^a - R\sim_i^a|^2 + \Sigma_j |R_j^a - R\sim_j^a|^2\} + \mu/2 .\{\Sigma_{i,j} s^a_{ij} |R_i^a - R_j^a|^2 + \Sigma_{ij} s^p_{ij} |R_i^p - R_j^p|^2\} + \tfrac{1}{2}. \Sigma_{i,j} w_{ij} |R_i^a - R_j^p|^2 \quad (3)$$

Where μ and λ are regularization parameters, s^a_{ij} is latent similarity between Apps and s^p_{ij} is latent similarity between permission.

5. CONCLUSION

This paper introduces a method which discovers a ranking fraud discovery system for mobile Apps. Initially, the proposed system describe that ranking fraud first happens in leading sessions and provided a processing method for mining leading. Then rating, ranking and review based evidences are used for detecting ranking fraud. An optimization based aggregation method is used to integrate all the evidences of the Mobile App. And evaluate the credibility of leading sessions from mobile Apps. Lastly, we developed a mobile App recommender system with security and privacy awareness. Specifically, without relying on any predefined risk functions, we designed a scalable and automatic approach for estimating the security risks of Mobile Apps.

ACKNOWLEDGEMENT

This work was supported in part by grants from the Dr. A.V. Ram Prasad, Principal of K.L.N. College of engineering and also supported by grants from DR.N. Lakshmi Narasimman, Professor & Head of Computer Science and Engineering (Project Guide), K.L.N. College of engineering, who had helped us during preparation and also provided valuable feedback for guidance.

REFERENCE

[1] Qi Liu, Enhong Chen, Hui Xiong, Chris H.Q. Ding, Jian Chen, "Enhancing Collaborative Filtering by User Interests Expansion via Personalized Ranking" IEEE Transactions on Systems, Man, and Cybernetics - Part B Vol. 42, Issue 1, pp. 218-233.2012

[2] Hengshu Zhu, Huanhuan Cao, Enhong Chen, Hui Xiong, Jilei Tian "Mobile App Classification with Enriched Contextual Information" IEEE Transactions on Mobile Computing, Vol. 13, Issue. 7, Pp 1550-1563, 2014

[3] H. Peng, C. Gates, B. Sarma, N. Liu, Y. Qi, R. Potharaju, C. Nita-Rotaru, I. Molloy "Using probabilistic generative models for ranking risks of android apps" ACM conference on Computer and Communication Security, Pp 939-948, 2010.

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

[5] 2014), [Online], Available: http://en.wikipedia.org/wiki/ information_retrieval

[6] (2015), [Online], Available: https://www.techinasia.com/ viral-photo-china-shows-manipulate-app-store-rankings- hard/

[7] Yong Ge, Guofei Jiang, Min Ding, Hui Xiong"Ranking Metric Anomaly in Invariant Networks" ACM Transactions on Knowledge Discovery from Data, Vol. 8, Issue. 2, 2014

[8] Ke Zhai, Jordan Boyd-Graber, "Online Latent Dirichlet Allocation with Infinite Vocabulary" Journal of Machine Learning Research,Vol. 28, Issue 1,Pp 561-569, 2013

[9] Hongyan liu, Jun he and yingqin gu, Hui xiong, Xiaoyong du, "Detecting and Tracking Topics and Events from Web Search Logs" ACM Transactions on Information Systems, Vol. 30, Issue. 4, 2012

[10] Ee-Peng Lim Viet-An Nguyen Nitin Jindal Bing Liu Hady W. Lauw, "Detecting Product Review Spammers using Rating Behaviors" ACM international conference on Information and knowledge management, pp 939-948, 2010.