Survey on Fraud Malware Detection in Google Play Store

Yuvaraj S[1], Dhinakaran P[2], DineshKumar S[3], JayaKumar K[4]

[1] *Assistant Professor, Department of Computer Science Engineering, Bannari Amman Institute of Technology, Sathyamagalam, India.*

[2, 3, 4] *UG Student, Department of Computer Science Engineering, Bannari Amman Institute of Technology, Sathyamangalam, India*

ABSTRACT - *The use of mobile devices including Tablets, Smart watch, and note books are increasing day by day. Android has the major share in the mobile application market. Android mobile applications become an easy target for the attackers because of its open source environment. Also user's ignorance the process of installing and usage of the apps. To identify fake and malware applications, all the previous methods focused on getting permission from the user and executing that particular mobile application. A malware detection framework that discovers and break traces left behind by fraudulent developers, to detect search rank fraud as well as malware in Google Play. The fraud app is detected by aggregating the three pieces of evidence such as ranking based, co-review based and rating based evidence. Finally aggregating all the activities of front running apps, it can be achieve certain accuracy in classifying benign standard datasets of malware, fraudulent and legitimate apps. Additionally, we apply incremental learning approach to characterize a large number of data sets. It combined effectively for all the evidences for fraud detection. To accurately locate the ranking fraud, there is a need to mining the active period's namely leading sessions, of mobile Apps.*

KEYWORDS - Mobile applications, Malware, Ranking, Rating, Google Play.

1. INTRODUCTION

Google play first releases its app in 2008.Since that it distributes applications to all the Android users. In Google Play Store, it provides services that user can discover the particular application, purchase those applications and install it on their mobile devices. Since Android is open source environment all the detail about the application users can be easily accessed by the application developers through Google play. In Google play 1.8 Million mobile applications are available and that is downloaded by over 25 billion users across the world. This leads to greater chance of installing malware to the applications that could affect users mobile devices. Google play store uses its own security system known as Bouncer system [6] to remove the malicious apps from its store. However, this method is not effective as testing some apps using virus tools many apps are found as malicious which are not detected by Bouncer system [6]. Fraudulent developers use search ranking algorithm to promote their apps to the top while searching. After downloading mobile applications from Google play users are asked to give the ratings and reviews about that particular downloaded applications. However fraudulent developers give fake ratings and reviews about their application promote their application to the top. There are two typical approaches used for detecting malware in Google Play. Thus are Static and Dynamic. The dynamic approach needs apps to be run in a secure environment to detect its benign. The static approach is not used as the need to give all types of attack in early stage itself but that is impossible as everyday attackers find the new way to inject malware on applications.

2. LITERATURE REVIEW

In paper [3] the author proposes the static method to detect the malware in mobile applications. In this system using reverse engineering concept the source code for the suspicious APK files. After that using structured mapping author builds the structure for the classes. Finally using data flow concept several patterns for the different type of threats has been created and use them to detect the malware in applications. Depending upon the number of threading pattern the effectiveness of this method is calculated.

In paper [1] the author proposed a new method to detect malware in mobile applications by examining the runtime behavior of that particular application in the mobile environment. The author proposes that unexpected behavior mobile app can vary from one application to other applications. Also, it varies from the environment of that particular application running on different devices. Using Xposed framework user can change the user and system application without modifying the application package(APK).Depend upon that user can set particular conditions to identify the malware in the mobile applications.

In paper [2] the author proposes some of modern machine learning algorithms to detect malware. For that these algorithms are applied to the metadata collected from the Google Play. While all of the existing methods for detecting algorithm focused on inherent characteristics of

the particular mobile app this gives a direct method to detect the applications. For the setup of the experiments the collected 25k data from Google Play. Developers update their applications in particular interval of days whereas fake applications could not be updated since its upload of the Google Play. All of these works focused on only linear models Future work may focus on non-liners models.

In paper [5] the author aims to protect the review spanners or spam reviews. The spammer may target only on the specific protect. After that, they gave fake reviews to that particular mobile app by creating the different account to review that account. The author proposes a novel based scoring method to detect every single review of the particular product. The author creates highly suspicious as a subset. By using web-based spammer evaluation software the fakeness of the review is calculated. After the completion of the evaluation, the result shows the effective to predict the fake reviews.

In paper [8] the authors have studied the problem of detecting hybrid shilling attacks on rating data. The proposed approach is based on the semi-supervised learning and can be used for trustworthy product recommendation. This paper presents a Hybrid Shilling Attack Detector, or HySAD for short, to tackle these problems. In particular, HySAD introduces MCRelief to select effective detection metrics, and Semi supervised Naive Bayes (SNBλ) to precisely separate Random-Filler model attackers and Average-Filler model attackers from normal users.

In paper [4], author proposed novel technique for computing a rank aggregation on the basis of matrix completion to avoid noise and incomplete data. Proposed method solves a structured matrix-completion problem over the space of skew-symmetric matrices. The author proves a recovery theorem detailing when proposed approach will work. They also perform a detailed evaluation of proposed approach with synthetic data and an anecdotal study with Netflix ratings. To find the solutions, they utilized the svp solver for matrix completion. Rank aggregation is combined with the structure of skew-symmetric matrices. Author applied for latest advances in the theory and algorithms of matrix completion to skew-symmetric matrices. Author enhanced existing algorithm for matrix completion to handle skew-symmetric data.

In paper[10], author reported a survey on Web spam detection, which comprehensively introduces the principles and algorithms in the literature. Indeed, the work of Web ranking spam detection is mainly based on the analysis of ranking principles of search engines, such as Page Rank and query term frequency. This is different from ranking fraud detection for mobile Apps. They 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 behavior, clicks, HTTP sessions. In turn, there is 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.

S. no	Paper Title	Techniques	Advantages	Drawbacks
1	Detecting Mobile Application Malicious Behaviors Based on Data Flow of Source Code.[3]	Static approaches to malware detection based on reverse code of the application.	User can define their own threat to identify malware.	Manually should define all threat patterns.
2	Mobile Malware Exposed[1]	XPosed Framework	Detect any type of Unexpected behavior.	Not Working for all type of malicious.
3	Android malware detection from Google Play meta-data: Selection of important features.[2]	Machine Learning Technique.	Detect all type of Threat Pattern.	Too Slow to detect Malware..
4	Detecting product review spammers using rating behaviors. [5]	A novel scheme to detect review spammers who try to influence review ratings on some target products or product groups.	After evaluation it proves that proposed ranking and supervised methods are effective in discovering spammers.	The scoring methods to compute the degree of spam for each reviewer.
5	Hysad: a semi supervised hybrid shilling attack detector for	Based on Historical Ranking.	Detect all type of Ranking Methods.	Not Effective.

	trustworthy product.[8]			
6	Rank aggregation via nuclear norm minimized. [4]	Novel technique for computing a rank aggregation on the basis of matrix completion to avoid noise and incomplete data	Proposed method solves a structured matrix completion problem over the space of skew symmetric matrices.	Matrix operation are difficult to perform.

Table 1: Comparative study on methods associated with fraud detection in Google play

3. PROPOSED SYSTEM:

It proposes malware detection framework system that effectively detects Google Play fraud and malware. To detect fraud and malware, we propose the incremental learning approach to characterize the dataset. We formulate the notion of review modeling by applying Porter stemmer algorithm. We use temporal session of review post times to identify suspicious review spikes received by apps; the application evidence such as rating, ranking and review evidence will be integrated by an unsupervised evidence-aggregation method for evaluating the credibility of leading sessions from mobile Apps. The malware detection framework is scalable and can be extended with other domain generated evidence for ranking fraud detection. When compared to other existing systems this method finds the better mobile app for the end user. Incremental learning approaches effectively characterize all category of app in Google Play. Also based on the review, rating and rank given by the user is also checked. User can review after they download that particular application using their account from app store.

3.1 ADVANTAGES:

- Detect fraud ranking in daily App leader board.
- Avoid ranking manipulation.
- Finds the better mobile app for the end user.
- Incremental learning approach effectively characterizes the large amount of app evidence details.
- It provides accurate aggregation when compared to our existing approach.

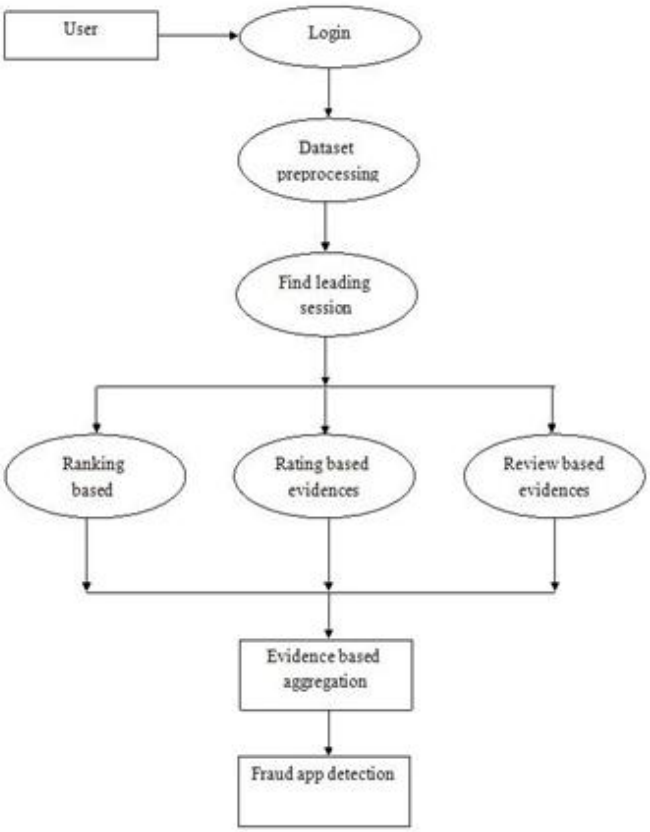

Fig 1: Incremental Learning approach

4. CONCLUSION:

In this project, we developed a fraud detection system for mobile Apps. Specifically, we first showed that fraud happened in leading sessions and provided a method for mining leading sessions for each App from its historical ranking records. We identified that for the detection of the rank ranking, rating, review based evidence are considered. Moreover, we proposed an optimization based aggregation method to integrate all the evidence for evaluating the credibility of leading sessions from mobile Apps. A unique perspective of this approach is that all the evidence can be modeled by statistical hypothesis tests, thus it is easy to be extended with other evidence 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 evidence and analyze the latent relationship among rating, review, and rankings. Moreover, we will extend our ranking fraud detection approach with other mobile App related services, such as mobile Apps recommendation, for enhancing user experience.

REFERENCES:

[1]Alaa Salman Imad H. Elhajj Ali Chehab Ayman Kayss, IEEE Mobile Malware Exposed.International Conference on Knowledge discovery and data mining, KDD'14 pages 978-983.

[2]Alfonso Munoz, Ignacio Mart ˜ ´ın, Antonio Guzman, Jos ´ e Alberto Hern ´ andez, IEEE Android malware detection from Google Play meta-data: Selection of important features.2015, pages,245-251.

[3]Chia-Mei Chen, Je-Ming Lin, Gu-Hsin Lai,IEEE Detecting Mobile Application Malicious Behaviors Based on Data Flow of Source Code.2014 International Conference on Trustworthy Systems and their Applications pp 95-109.

[4]D. F. Gleich and L.-h. Lim. Rank aggregation via nuclear norm minimization. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '11, pages 60–68, 2011.Y.T. Yu, M.F. Lau, "A comparison of MC/DC, MUMCUT and several other coverage criteria for logical decisions", Journal of Systems and Software, 2005, in press.

[5]E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw. Detecting product review spammers using rating behaviors. In Proceedings of the 19th ACM international conference on Information and knowledge management, CIKM '10, pages 939–948, 2010.

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

[7]J. Oberheide and C. Miller, "Dissecting the Android Bouncer," presented at the SummerCon2012, New York, NY, USA, 2012.

[8]K.Shi and K.Ali. Getjar mobile application recommendations with very sparse datasets. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '12, pages 204–212, 2012.

[9]J.Kivinen and M. K. Warmuth, "Additive versus exponentiated gradient updates for linear prediction," in Proc. 27th Annu. ACM Symp. Theory Comput., 1995, pp. 209–218.

[10]N. Spirin and J. Han. Survey on web spam detection: principles and algorithms. SIGKDD Explor. Newsl.,13 (2):50–64,May2012.