Eleyon SOUTH ASIAN JOURNAL OF ENGINEERING AND TECHNOLOGY

Full Length Article

SearchRank Fraud and Malware Detection in Google play using Finer Cluster process

S. Nivethini[2]*, B. Logendhiran*[2]*, B. Anjitha*[2]*, V. Kavitha*[1]

[1] *Assistant professor, Department of Information Technology, Nandha College of Technology, Erode 638052 ,Tamilnadu ,India*

[2] *UG Scholar Department of Information Technology, Nandha College of Technology, Erode- 638052, Tamilnadu, India*

[2] *UG Scholar Department of Information Technology, Nandha College of Technology, Erode- 638052, Tamilnadu, India*

[2] *UG Scholar Department of Information Technology, Nandha College of Technology, Erode- 638052, Tamilnadu, India*

[2] *UG Scholar Department of Information Technology, Nandha College of Technology, Erode- 638052, Tamilnadu, India*

***Corresponding Author**
kavithamebit@gmail.com
(V.Kavitha)
Tel.: +91 9791209971

ABSTRACT: The commercial success of Android app markets such as Google Play and the incentive model they offer to popular apps, make them appealing targets for fraudulent and malicious behaviours. Some fraudulent developers deceptively boost the search rank and popularity of their apps while malicious developers use app markets as a launch pad for their malware. In this project, introduce FairPlay , a novel system that discovers and leverages traces left behind by fraudsters, to detect both malware and apps subjected to search rank fraud. This project shows that an adversary can successfully infer a victim's vertex identity and community identity by the knowledge of degrees within a time period. The project also includes a new supervised clustering algorithm to find groups of data (coarse and finer cluster). It directly incorporates the information of sample categories into the fraud clustering process.

Keywords: Tweets collection, Co-Review Graph, Find Cliques, Threshold value

1 Introduction

1.1 Social Network

A social networking service (SNS) is a platfo m to build social networks or social relations among people who share similar interests, activities, backgrounds or real-life connections. A social network service consists of a representation of each user often a profile, his or her social links, and a variety of additional services. Social network sites are web-based services that allow individuals to create a public profile, create a list of users with whom to share connections, and view and cross the connections within the system.

The Most social network services are web-based and provide means for users to interact over the Internet, such as e-mail and instant messaging. Social network sites are varied and they incorporate new information and communication tools such as mobile connectivity, photo, video, sharing. The Online community services are sometimes considered a social network service, though in a broader sense, social network service usually means an networking sites allow users to share ideas, pictures, individual-centered service whereas online community services are group-centered. Social posts, activities, events, and interests with people in their network.

1.2 Security Management

Security management for networks is different for all kinds of situations. A home or small office may only require basic security while large businesses may require high-maintenance and advanced software and hardware to prevent malicious attacks from hacking and spamming. An attack can be perpetrated by an insider or from outside the organization An "inside attack" is an attack initiated by an entity inside the security perimeter an "insider", an entity that is authorized to access system resources but uses them in a way not approved by those who granted the authorization. An "outside attack" is initiated from outside the perimeter, by an unauthorized or illegitimate user of the system "outsider".

1.3 Objectives

The main objectives of the FairPlay are

1. To automatically detect malicious and fraudulent apps.
2. To correlate review activities and uniquely combines detected review relations with linguistic and behavioural signals.
3. To discover and leverage traces left behind by fraudsters.
4. To detect both malware and apps subjected to search rank fraud.

The achieve the main goal, the specific objectives required are

1. To create a The Co-Review Graph (CoReG) that identifies apps reviewed in a contiguous time window by groups of users with significantly overlapping review histories.
2. To propose review feedbacks approach which exploits feedback left by genuine reviewers?
3. To prepare clique from the Co-Review graph so that most related fraudulent users are found out.

2. System Analysis

2.1 Existing System

The existing system seeks to identify both malware and search rank fraud subjects in Google Play using FairPlay. This combination is not subjective, it speculate that malicious developers resort to search rank fraud to boost the impact of their malware. The proposed system is built on the observation that fraudulent and malicious behaviors leave behind telltale signs on app markets. FairPlay is a Fraud and Malware Detection Approach which formulate the notion of co-review graphs to model reviewing relations between users. The temporal dimensions of review post times are used to identify suspicious review spikes received by apps. The linguistic and behavioral information is used to detect genuine reviews from which and then extract user- identified fraud and malware indicators. In the existing system, The Co-Review Graph (CoReG) module identifies apps reviewed in a contiguous time window by groups of users with significantly overlapping review histories. The Review Feedback (RF) exploits feedback left by genuine reviewers, while the Inter Review Relation (IRR) leverages relations between reviews, ratings and install counts.

2.2 Drawbacks

2. It does not consider the protection of vertex and community identities of individuals in a dynamic network.
3. A privacy model for protecting multi-community identity is not carried out.
4. It is identifying both malware and search rank fraud subjects alone not privacy breaches.

2.3 Proposed System

Proposed system includes a new supervised clustering algorithm is proposed to find groups of fraud. It directly incorporates the information of sample categories into the fraud clustering process. A new quantitative measure is introduced that incorporates the information of sample categories to measure the similarity between users. The proposed algorithm is based on measuring the similarity between users using the new quantitative measure. So redundancy among the fraud is removed. Less dense nodes in the graph is removed so users who rating in minimum amount are not treated as fraud users.

2.4 Advantage

1. Number of clusters is prepared and relevance among the fraud is filtered such that coarse as well as finer cluster is prepared.
2. Cliques preparation correctly identifies fraud users.
3. Densely connected fraud users are also tracked in graphs.
4. Less dense nodes in the graph is removed so users who rating in minimum amount are not treated as fraud users.

3. Project Description

3.1 Project Definition

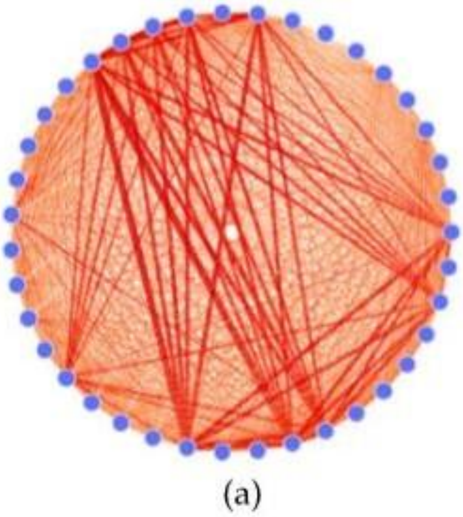
Commercial success of Android app markets like Google Play] and the incentive model they offer to popularize apps, make them appealing targets for malicious and fraudulent behaviors. Some fraudulent developers deceptively boost search rank and popularity of their apps (e.g., through fake reviews and bogus installation counts) while malicious developers use app markets as a launch pad for their malware. The motivation for such behaviors is impact: app popularity surges translate into financial benefits and expedited malware proliferation. The main problem is to detect malicious and fraudulent apps. Hence if a system that leverages the above observations to efficiently detect Google Play fraud and malware, then it will be helpful. So the project introduces FairPlay, a system to automatically detect malicious and fraudulent apps.

3.2 Fairplay

FairPlay organizes the feedbacks given by users and preprocesses the reviews. Then the CoReview Graph is being constructed. This CR Graph exploits the observation that fraudsters who control will re-use them across multiple jobs. Its goal is then to detect sub-sets of an app's reviewers that have performed significant common review activities in the past. In the following, we describe the co-review graph concept, formally present the weighted maximal clique enumeration problem, then introduce an efficient heuristic that leverages natural limitations in the behaviors of fraudsters.

3. 3 CoreviewGraph

Co-Review Graphs: Let the co-review graph of an app, a graph where nodes correspond to user accounts who reviewed the app, and undirected edges have a weight that indicates the number of apps reviewed in common by the edge's endpoint users.



3.1 Clique detection

Fig. 3.1 shows the co-review clique of one of the seed fraud apps. The clique contains 37 accounts (names hidden for privacy) that reviewed the app. The edge weights are suspiciously high: any two of the 37 accounts reviewed at least 115 apps and up to 164 apps in common.

The co-review graph concept naturally identifies user accounts with significant past review activities. The Weighted Maximal Clique Enumeration Problem. Let G = (V, E) be a graph, where V denotes the sets of vertices of the graph, and E denotes the set of edges. Let w be a weight function, w : E -> R that assigns a weight to each edge of G. Given a vertex sub-set U V, we use G[U] to denote the sub-graph of G induced by U. A vertex sub-set U is called a clique if any two vertices in U are connected by an edge in E. We say that U is a maximal clique if no other clique of G contains U. The weighted maximal clique enumeration problem takes as input a graph G and returns the set of maximal cliques of G.

Maximal clique enumeration algorithms applied to co- review graphs are not ideal to solve the problem of identifying sub-sets of an app's reviewers with significant past common reviews. First, fraudsters may not consistently use (or may even purposefully avoid using) all their accounts across all fraud jobs that they perform. In addition, Google Play provides incomplete information (up to 4,000 reviews per app, may also detect and filter fraud). Since edge information may be incomplete, original cliques may now also be incomplete. To address this problem, we "relax" the clique requirement and focus instead of pseudo-cliques:

3.4 PseudoCliqueFinder(PCF)Algorithm

A PCF (Pseudo Clique Finder), an algorithm is proposed that exploits the observation that fraudsters hired to review an app are likely to post those reviews within relatively short time intervals (e.g., days). PCF (see Algorithm 1), takes as input the set of the reviews of an app, organized by days, and a threshold value u. PCF outputs a set of identified pseudo-cliques with r u, that were formed during contiguous time frames.

For each day when the app has received a review (line 1), PCF finds the day's most promising pseudo-clique (lines 3 and 12 -22): start with each review, then greedily add other reviews to a candidate pseudo-clique; keep the pseudo clique (of the day) with the highest density. With that "work-in-progress" pseudo-clique, move on to the next day (line 5): greedily add other reviews while the weighted density of the new pseudo-clique equals or exceeds u (lines 6 and 23 - 27). When no new nodes have been added to the work-in- progress pseudo-clique (line 8), we add the pseudo- clique to the output (line 9), then move to the next day (line 1). The greedy choice (get Max Density Gain, not depicted in Algorithm 1) picks the review not yet in the work-in-progress pseudo-clique, whose writer has written the most apps in common with reviewers already in the pseudo-clique.Fig.1 illustrates the output of PCF for several values.

3.5 Module Description

The following modules are present in the project.

- Tweets Collection for reviews.
- Co-Review Graph Construction.
- Finding Cliques to get fraud users.
- Remove nodes with edge weights

- below threshold so normal users are

3.5.1 Tweets Collection for reviews

Using twitter package and search twitter function, the tweets are downloaded and preprocessed.

Stop word removal, punctuation removal, unicode character removal are carried out. Key Terms are filtered such that first 50 more occurrence words are taken. Then unique users in the tweet are also found out.

3.5.2 Co-Review Graph Construction

From unique users in the tweet are found out. Same Key word present in two topics of two different users are found, then two nodes and one edge is formed in the graph. Thus the full graph is constructed. During edge addition, co-occurrence count is also found out and set as edge weight.

3.5.3 Finding Cliques

From the full graph constructed, cliques are found out with minimum 5 nodes in them. These cliques denote the users who are densely connected. These users are treated as fraud users.

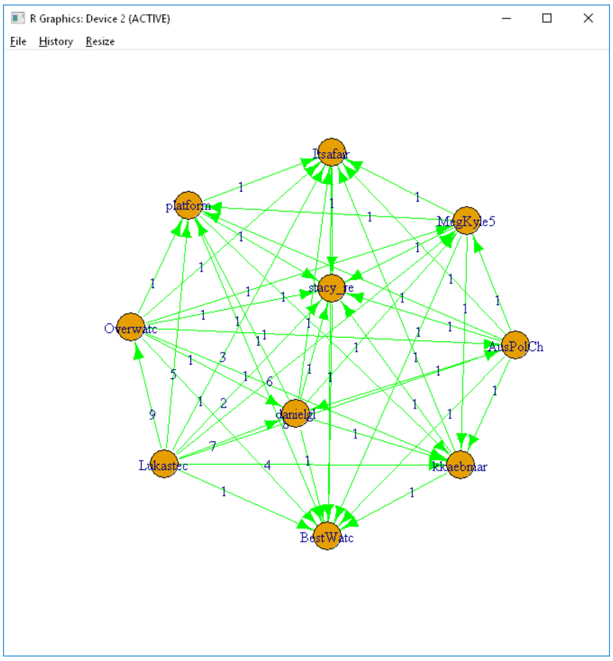
3.5.4 Remove Nodes with Edge Weights below Threshold so Normal users are treated as Non-Fraud users

One nodes, all edges are taken. If all the edge weights are below the given threshold values, it means the user is giving rating less times only. The user is treated as normal user.

4 Conclusion

Some fraudulent developers deceptively boost the search rank and popularity of their apps (e.g., through fake reviews and bogus installation counts), while malicious developers use app markets as a launch pad for their malware. The motivation for such behaviors is impact: app popularity surges translate into financial benefits and expedited malware proliferation. This project seeks to identify both malware and search rank fraud subjects in Google Play. This combination is not arbitrary: we posit that malicious developers resort to search rank fraud to boost the impact of their malware. Unlike existing solutions, this project builds this work on the observation that fraudulent and malicious behaviors leave behind telltale signs on app markets. The project has introduced FairPlay, a system to detect both fraudulent and malware Google Play apps. The experiments on the twitter posts, have shown that a high percentage of fraud users are found. In addition, it showed FairPlay's ability to discover non-fraud users also.

Clique detection



References

[1] I. Burguera, U. Zurutuza, and S. Nadjm-Tehrani , "Crowdroid: Behavior-based Malware detection system for Android," in Proc. ACM SPSM, 2011, pp. 15–26.

[2] A. Shabtai, U. Kanonov, Y. Elovici, C. Glezer, and Y. Weiss, "Andromaly: A behavioral malware detection framework for Androiddevices, "Intell.Inform. Syst.,vol.38, no.1, pp.161–190,2012.

[3] M. Grace, Y. Zhou, Q. Zhang, S. Zou, and X. Jiang, "RiskRanker: Scalable and accurate zero- day Android malware detection," in Proc. ACM MobiSys, 2012, pp. 281–294.

[4] H. Peng, et al., "Using probabilistic generative models for ranking risks of Android Apps," in Proc. ACM Conf. Comput. Commun. Secur., 2012, pp. 241–252.

[5] S. Yerima, S. Sezer, and I. Muttik, "Android Malware detection using parallel machine learning classifiers," in Proc. NGMAST, Sep. 2014, pp. 37–42.

[6] J. Sahs and L. Khan, "A machine learning approach to Android malware detection," in Proc. Eur. Intell. Secur. Inf. Conf., 2012, pp. 141–147.

[7] B. Sanz, I. Santos, C. Laorden, X. Ugarte-Pedrero, P. G. Bringas, and G. Alvarez, "Puma: Permission usage to detect malware in android," in Proc. Int. Joint Conf. CISIS12-ICEUTE' 12- SOCO' Special Sessions, 2013, pp. 289–298.

[8] J. Ye and L. Akoglu, "Discovering opinion spammer groups by network footprints," in Machine Learning and Knowledge Discovery in Databases. Berlin, Germany: Springer, 2015, pp. 267–282.

[9] D.H.Chau,C.Nachenberg,J.Wilhelm,A.Wright,andC.Faloutsos, "Polonium: Tera-scale graph mining and inference for malware detection," inProc. SIAMInt. Conf .DataMining, 2011 Art.no.12.

[10] A. Tamersoy, K. Roundy, and D. H. Chau, "Guilt by association: Large scale malware detection by mining file-relation graphs," in Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2014, pp. 1524–1533. [Online]. Available: http://doi.acm.org/10.1145/2623330.2623342