Ranking Fraud Detection for Mobile Applications

A Nikhila[1], Deepashree C[2], Jayanthi R C[3], Jalaja G[4]

[123] Department of Computer Science and Engineering, BNM Institute of Technology, Karnataka, India
[4]Associate Professor, Department of Computer Science and Engineering, BNM Institute of Technology, Karnataka, India

---

Abstract - *The mobile industry is growing rapidly, subsequently the number of mobile apps coming in the market is also increasing. Google Play and Apple are app stores where apps can be downloaded by either paying or they can be downloaded free of cost. Rank of an app is important, as high ranked apps get noticed by customers more easily than those ranked low. In order to get a high rank some developers are resorting to fraudulent means in order to increase their app's rank. Hence a ranking fraud detection system is required to detect rankings obtained through fraudulent means. We design and develop a ranking fraud detection system to detect fraud ranks. The system takes app's details and reviews and stores them in a database. Using these information, the reviews are preprocessed, and sentimental analysis is performed. The results obtained are compared with the rating of the app from which raking fraud is detected.*

Key Words: Ranking fraud detection, Reviews, Ranking, Data mining, Mobile applications

1. INTRODUCTION

Mobile users are increasing day by day and with the invention of applications in smartphones made lives easy. Many applications make our daily lives easy such as bank payments, online shopping applications etc., The increase in the usage of such applications made many different applications. Some applications are good applications and some of them are fake. There is no harm in installing good applications where as the fraud applications may reduce the performance in the mobile or delete some important data. This kind of applications are made by fraud application makers and create fraud ratings for that applications. So, there is a vital need of detecting such fraud ratings. The importance of preventing ranking fraud has been widely recognized, there is limited understanding and research in this area. To this end, in this system, we provide a holistic view of ranking fraud and propose a ranking fraud detection system for mobile Apps. This system conducts sentimental analysis on the reviews of the mobile app and it calculate the number of positive and negative reviews, fetch the ranking of the mobile app, do a ranking analysis and determine whether there is any ranking fraud happen or not. This system uses machine learning technique by which positive and negative new words are appended automatically to the keyword database. Furthermore, IP address blocking facility is added to the system.

This paper briefly describes our ranking fraud detection system which detects whether the fraud had been occurred in the ranking of that application or not. It furthermore describes about IP blocking system which we used to filter reviews.

2. LITERATURE SURVEY

In paper [1] they developed a ranking fraud detection system for mobile apps. They observed that mobile apps are not always ranked high in leaderboard but only in some leading events which collectively form different leading sessions. Fraudulent apps often have different ranking patterns in each leading session compared with normal apps. Since an app's ranking can be affected by app developer's reputation and marketing campaigns like limited time discount they also collected fraud evidences based on apps rating and review history. The evidences were extracted by modeling apps' ranking, rating and review behaviors through statistical hypotheses tests. Unsupervised evidence-aggregation method was used to integrate the three types of evidences. The proposed system was evaluated system with real-world app data collected from the Apple's App store for a long period, of about two years. Experimental results obtained showed the effectiveness of the proposed system, the scalability of the detection algorithm as well as some regularity of ranking fraud activities.

In paper [2], they proposed a sequential approach based on Hidden Markov Model(HMM) to model the heterogeneous popularity information for mobile apps including chart ranking, user rating and review topics extracted from user reviews. It also aims to provide a comprehensive modeling of popularity information for mobile app services. The popularity information is modelled in terms of different transitions of different popularity states, which indicate the latent semantics and relationships of popularity observations. Popularity information include chart rankings, user ratings and user reviews. The PHMM model can be used to build other applications like trend based mobile app recommendation, rating and review spam detection and ranking fraud detection.

In paper [3], The proposed model in this system first preprocess the reviews and tokenizes the words. It then converts words into its root word and removes those words which are not required. Scores are given to the words extracted followed by which scores are given to emoticons. All possible emoticons are stored in a dictionary along with its score. The overall score is used to classify if the reviews are positive or negative. Evidences were then aggregated,

and results obtained told if the app had a ranking fraud or not.

In paper [4], they studied about malware and search rank fraud subjects in Google Play. They this by discovering and leveraging traces left behind by fraudsters. To detect both

malware and apps subjected to search rank fraud they built a system called FairPlay. Co-review graphs were used to model reviewing relations between users. Temporal dimensions of review post times were used to identify suspicious review spikes received by apps. Apps with unbalanced review, rating and install counts as well as apps with permission request ramps were identified. Instead of analyzing app executables, linguistic and behavioral information was used to detect genuine reviews from which user identified fraud and malware were extracted.

In paper [5], they performed a detailed temporal analysis on two datasets that they collected from the Google Play Store, and other newly released apps. They have tried to shed light on the dynamics of Google Play. They developed iMarket, a prototype app market crawling system to collect data. They developed IP blocking system which blocks the reviews from same IP after standard count of reviews from that IP address. The longitudinal study of Google Play app metadata provides unique information that is not available through the standard approach of capturing data. Features extracted from longitudinal app analysis like permission price, update and download count changes provide insights into fraudulent app promotion and malware indicator behavior. For instance, spikes in the number of positive or negative reviews and the number of downloads received by an app can indicate app search optimization campaigns launched by fraudsters who were recruited through crowdsourcing sites like Freelancer. Frequent substantial app updates may indicate denial of service attacks, while permission changes can indicate benign apps turning malicious.

3. EXISTING SYSTEM

The system model incorporates substances: Mobile Apps, historical record, mining session, three evidences that are ranking, rating and review.

Mining leading session: This method calculates the mining leading session from historical records of mobile apps of apps industry. Ranking fraud does not always happen in the whole life cycle of an App, but only in some leading events, which form different leading sessions. Mining method is applied on these leading sessions examining ranking, rating and review behaviors.

Evidence aggregation: Aggregation method is applied on these three evidences. If false, mobile app is the output of this aggregation system then this app is prevented from being recommended to the user.

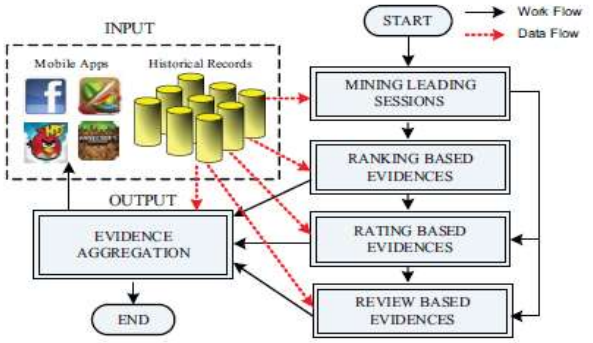

Fig -1: Existing model for ranking fraud detection for mobile apps

4. PROPOSED SYSTEM

In this project, we use effective algorithms for Auto Inclusion of Sensitive Word(Self-learning), Keyword identification, Preprocessing algorithm, Keyword adaptive sentiment classification and Auto Inclusion of Sensitive Word. These algorithms build the system whose framework is mentioned below.

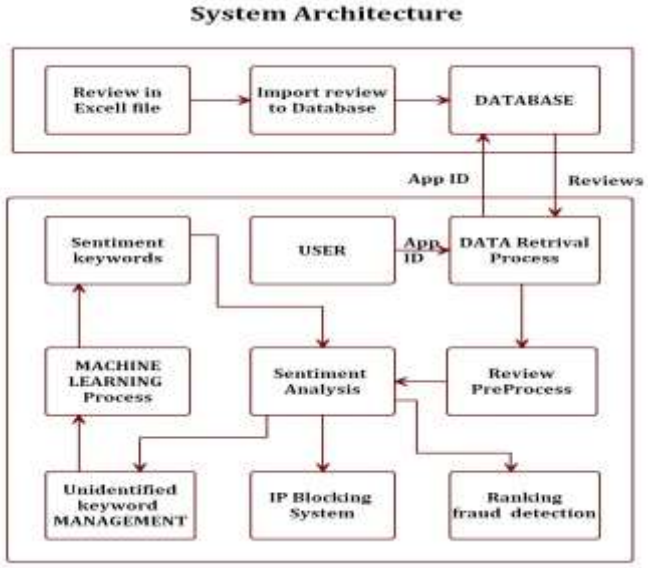

Fig -2: The Framework of ranking fraud detection system for mobile Apps

5. ALGORITHMS

5.1 IP FILTERING ALGORITHM

Step 1: Let n be the number of reviews

Step 2: Keep a count for each unique IP address

Step 3: for i=0 to n

Step 4: If count < 5 for a IP address, block the IP address

Step 5: If i=n go to step 6, else go to step 4

Step 6: End of loop

Step 7: Stop

This algorithm is used to detect and remove fraud reviews given by bot farms. Every user who gives a review has an IP address from which he gives reviews. Those reviews given by bot farms will have same IP address, so in this algorithm those reviews which have same IP address would be fake reviews and hence are deleted since those reviews will not be considered.

5.2 PREPROCESSING ALGORITHM

Step 1: Let n be the number of reviews

Step 2: for i=0 to n

Step 3: Remove unnecessary words from reviews

Step 4: Remove Hyperlinks from reviews

Step 5: Remove Special characters

Step 6: If i=n go to Step 7 else go to step 3

Step 7: End of loop

Step 8: Stop

The preprocessing algorithm is used first before sentimental analysis to get the keywords necessary to perform sentiment analysis. Taking n reviews, for each review the unnecessary words like stop words which are irrelevant to perform sentiment analysis, hyperlinks and special characters are removed in order to make analyzing easier and simple. This is done for all n reviews.

5.3 AUTO INCLUSION OF SENTIMENT WORD

Step 1: Let n be the number of reviews

Step 2: Extract sentiment words

Step 3: Initialize sentiment words based on whether they are positive, negative or neutral

words

Step 4: Repeat Step 2 and Step 3 for i=1 to n

Step 5: Stop

This algorithm is used in order to make the system learn to score sentiment words based

on whether it is positive, negative or neutral. Taking n reviews, the sentiment keywords are first extracted for each review. The extracted sentiment words are then initialized as positive, negative or neutral words. This process is repeated till all n reviews are read.

5.4 KEYWORD ADAPTIVE SENTIMENT CLASSIFICATION

Step 1: Let n be the number of reviews

Step 2: for i=0 to n

Step 3: Count the number of positive, negative, neutral count of the sentiment keyword in a review.

Step 4: Based on the overall count give sentiment process to tell if app is good or bad.

Step 5: if i=n go to step 6 else go to step 3

Step 6: End of loop

Step 7: Stop

This algorithm is used to tell if a review is positive, negative or neutral based on keywords in the review. Taking n reviews, count the number of positive, negative and neutral words. The overall count tells whether the app is whether the app.

6. DATASET

There are two datasets used in this project. One of the dataset consists of the app's URL, app's name and its overall rating. The other dataset consists of reviews and the IP address of the app for which the reviews was written for. Each app has only one URL, name and rating while it has many reviews. Hence, the reviews are kept as a separate dataset along with the app's IP address in order to backtrack for which app the review was written for. Snapshot of the two datasets are as shown in Figures below.

The dataset1 contains the following fields:

• App_URL: App's web address

• App_Name: Name of the app

• Rating: Overall rating of the app The dataset2 consists of the following fields:

• Reviews: User's experience of the app

• ip address: The IP address of the app for which the review was written for

	A	B	C
1	App_Url	App_Name	Ratings
2	com.outfit7.gingersbirthdayfree	Talking_Ginger	1
3	com.tinder	Tinder	2
4	com.okcupid.okcupid	okcupid_Dating	5
5	com.quanticapps.quranandroid	Quran_pro_Muslim	1
6	com.myntra.android	myntra	5
7	com.myairtelapp	MyAirtelApp	4
8	com.jabong.android	jabong	3
9	com.disney.WMW	disney	4
10	com.gobit.burgershop	Burger_Shop_Freee	2
11	com.360booster	360_booster	1

Fig -3: Snapshot of dataset1

	A	B
1	Reviews	ip address
2	good app	124.65.23.4
3	beautiful app	221.90.87.45
4	hardly works unless u pay for it	213.56.23.47
5	very broken!	112.34.90.56
6	scam for sure	78.234.45.67
7	best one!	123.54.143.67
8	just amazing	213.56.23.47
9	really got irritated	231.54.121.90
10	king of all apps	125.64.234.98
11	worst one ever used	234.65.90.12
12	thank u for this app	122.156.92.14
13	not that good	124.23.53.87
14	bad one	213.56.23.47
15	very interesting	121.34.56.7
16	waste of memory	145.76.23.1
17	never use it	65.231.67.90
18	worst one	213.56.23.47
19	disgusting	123.120.34.46
20	great work	222.65.90.12

Fig -4: Snapshot of dataset 2

7. RESULTS

The snapshots show the working of the ranking fraud detection system. Initially when the system is started, user is shown to user login page. In case user does not have an account, he can register by clicking the link User Registration. If user has an account, he can enter his username and password to access his account. After logging in, user is directed to his account, where he can see his profile, upload apps and reviews, change password, perform perception analysis, see application status, graph of the analysis of the apps and logout when he is done.

The user login page is as shown in Fig- 5. User can register if he is new or else login in by entering his username and password. After user has successfully logged in, he his taken to his account where the initial display is the user's profile.

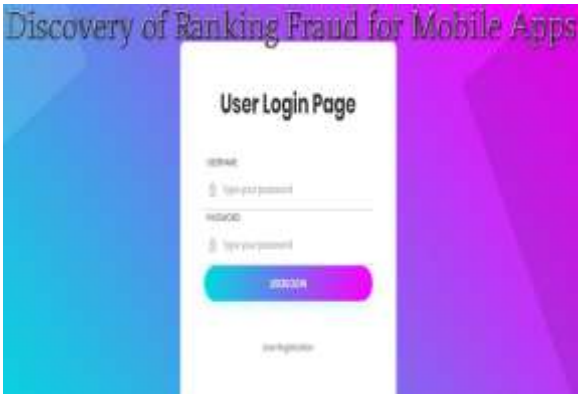

Fig-5: User login page

Fig-6: User profile

User can upload app and reviews by selecting the Upload Apps and Upload Reviews tab respectively.

Fig-7: Upload review

Fig-8: Perception analysis

After uploading the apps and their review, user can perform perception analysis, to analyze the reviews, whether they are positive, negative or neutral based on keywords obtained.

After performing perception analysis, user can view the results of the analysis by selecting the application status tab and giving app's name as the input.

Fig-9: Application status result for no ranking fraud

Fig-10: Application status result for ranking fraud undecidable.

Application status tells the total number of reviews considered, count of positive, negative and neutral reviews. From this the application tells if there is a fraud in ranking by taking the rating of the app into consideration and the result of the review analysis. The results can be that the app has a ranking fraud or not or it can't be decided. As apps are analyzed for ranking fraud, a graph is drawn which tells how much positive reviews the app has in percentage.

Fig-11: Graph representing percentage of perception

8. CONCLUSION

This project presented a Ranking Fraud Detection system for mobile application. A user who has an account can check for rank fraud in an application. Users can upload a list of applications for which fraud is to be detected as well as reviews of those apps. Perceptual analysis can be done for the reviews of an app, which analysis if a review is positive, negative or neutral. Based on perception analysis and the ratings of the app, the system can detect ranking fraud.

9. FUTURE WORK

The future work of this project includes taking other evidences like download count and app developers reputation to increase the accuracy of the system and taking data directly from app stores instead of uploading them as files.

REFERENCES

[1] Honshu Zhu, Hui Xiong, Senior Member, IEEE, Yong Ge, and Enhong Chen, Senior Member, IEEE "Discovery of Ranking Fraud for Mobile Apps" IEEE Transactions On Knowledge And Data Engineering, Vol. 27, No. 1, January 2015.

[2] Raju Dara, Pasupuleti Anil sai , International Journal of Modern Computer Science and Applications (IJMCSA) , Dept.of CSE, JNTUH, "Leading Session Mining and Evidence Based Approach for Ranking Fraud Detection of Mobile Applications", Volume No.-5, Issue No.-1, January, 2017

[3] V. Pingale, L. Kuhile, P. Phapale, P. Sapkal, S. Jaiswal. " Fraud detection & prevention of mobile apps using optimal aggregation method." International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE). 2016; 6(3), 491-497.

[4] Mahmudur Rahman, Mizanur Rahman, Bogdan Carbunar and Duen Horng Chau. "Search Rank Fraud and Malware Detection in Google Play." IEEE Transactions on knowledge and data engineering, vol. 29, no. 6, June 2017

[5] Rahul Potharaju, Mizanur Rahman, and Bogdan Carbunar "A Longitudinal Study of Google Play". IEEE Transactions on computational social systems, vol. 4, no. 3, September 2017.

[6] https://www.wikipedia.org/datamining

[7] https://ieeexplore.ieee.org/