ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Vol. 4, Issue 4, April 2016

Ranking Fraud Detection for Mobile Apps using Evidence Aggregation and Humming Bird algorithm

S.Kalaiarasi[1], Swetha Ganesh[2], Praveena.C.H[2], Vaishali.T[2]
[1]Assistant Professor II/IT, Prathyusha Engineering College, Thiruvallur-602025, India.
[2]Final year/IT, Prathyusha Engineering College, Thiruvallur-602025,India.

ABSTRACT: There are a lot a number of apps that are increasing every day over the past few years. The owners also resort to shady and fraudulent activities to increase the ranking of the apps in the popularity list. There is limited understanding in this area though the prevention of fraud has been widely is recognized. In Proposed we are predicting how many users using the particular apps based on their downloading the limitation then we are providing all kind of supportable apps like Android, Windows, IOS, and Symbian. In this paper, we provide a view of ranking fraud for mobile Apps. The users are provided a limitation of using the apps. The user can download the apps by providing the secret key which is provided by the admin. And when the users are trying to misuse the apps by downloading it a number of times, the user information is send to the Admin. We are also predicting how many users are using the particular App. Also, in the existing system even if the user views the app details, the app ranking is being increased. But in this system, only if the user downloading the App will increase the ranking of the particular app. The usage of apps can also be tracked using the leading apps and the graph of the particular app can also be tracked.

KEYWORDS: Fraud detection, humming bird algorithm, evidence aggregation, ranking fraud, secret key.

I. INTRODUCTION

There are a lot of mobile apps that is growing at a fast and steady rate over the past years. As a survey suggests there are around 1.6 million apps in the android market.

There is even a leader board to specify the apps that occupy the top position among the list of Apps. A app which occupies a higher position on the leader board leads to a large number of downloads and which in turn leads to a large revenue. Therefore, the developers of the App resort to various fraudulent means to bring their app to the highest position in the leader board. In order to have their particular app ranked in highest of the other apps, they try various measures to bring their apps in the leader board. This is usually done by using “bot farms” or “human water armies” to increase the Apps in a very short time. A survey states that when an App is promoted with the help of these shady means it could be propelled from number 1800 to the top 25 in any leader board and many number of users can be acquired within a couple of days. It causes a great danger to the mobile App industry. Even if there are many articles regarding the spam detection and mobile apps recommendation in the literature, the prospect of ranking fraud has been still not explored. In this paper, we propose to detect ranking fraud in mobile apps. In the current scenario, users tend to judge the usefulness and the effectiveness of the App by viewing the reviews and the ranking of the particular App This also can prove to be a major concern as it can also provide a false review to the people viewing the app details This can also be caused by malicious users who tend to post negative reviews intentionally to decrease the app ratings and to provide a negative image of the particular App. Firstly, when the user tries to download a particular app he is provided with a secret key by the Admin. Only when the secret key is entered the app can be downloaded by the user. Secondly, when the user tries to download a particular App many number of times, he is blocked and the user details are sent to the Admin. The Admin blocks the user from any further downloading. Thirdly, only when the user downloads the App the ranking of the particular app is increased. Experimental results show the effectiveness of the proposed system and the prevention of the fraud activities that take place in the life cycle of an Application.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

II. LITERATURE SURVEY

In this section, the previous work regarding this particular theme was collaborated and studied. An empirical study has been conducted investigating the relationship between the performance of an aspect based language model in terms of perplexity and the corresponding information retrieval performance obtained in [1], but this did not provide superior performance. The study over all the relationships and models were omitted due to space constraints. This paper identified the relationship between the language model perplexity and IR precision call measures. However, this model did not provide superior IR performance. Given the dynamic nature of the Web, where data sources are constantly changing, it is crucial to automatically discover these resources [2]. In this paper, a new crawling strategy is proposed to automatically locate hidden-Web databases which aim to achieve a balance between the two conflicting requirements of this problem: the need to perform a broad search while at the same time. The proposed strategy does that by focusing the crawl on a given topic; by judiciously choosing links to follow within a topic and by employing appropriate stopping criteria. However, this model was not possible to manually check all the forms that are being retrieved. In [3], the fraud detection system for mobile apps has been studied and it is provided a holistic view. The three types of evidences namely the ranking, rating and the review were analysed and aggregated to discover the fraud measures. The leading sessions and the leading events of the app were studied using the mining leading sessions algorithm. But, this model failed to explain the relationship between the three evidences and it also failed to provide a secure means of downloading and using the app. In [4], it proposed Facebook's Rigorous Application Evaluator (FRAppE). It failed to recommend to the website the hackers.

III. EVIDENCE AGGREGATION

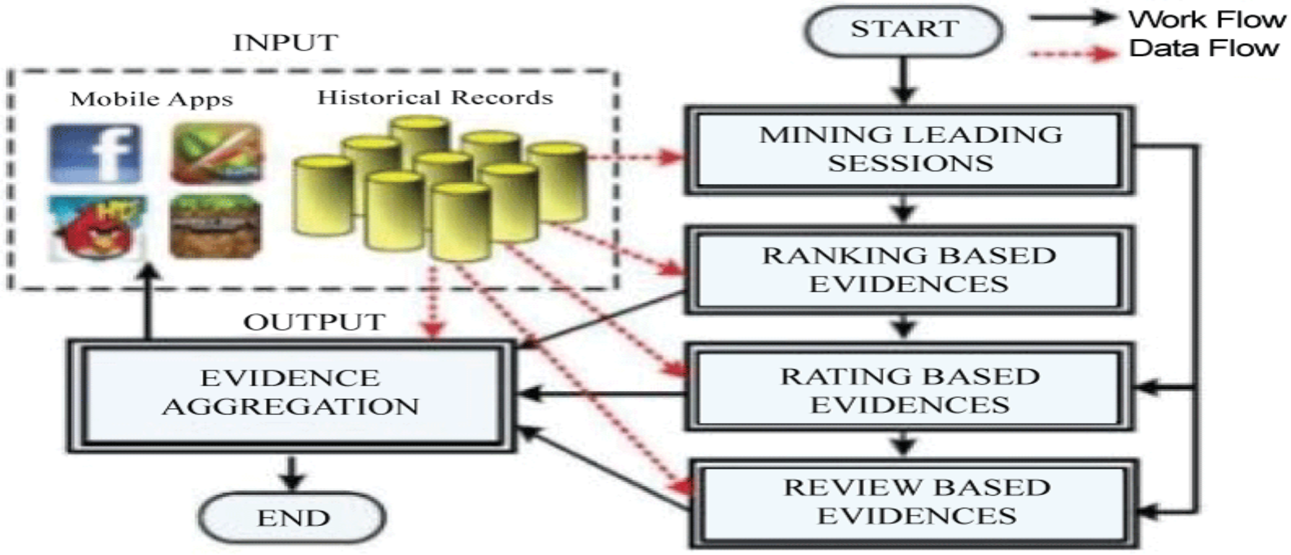

Figure 1: Evidence aggregation model.

To This represents as to how the evidences that are collected are being used to locate the ranking fraud in mobile Apps. The three evidences namely the ranking, rating and the review are identified and are aggregated using the evidence aggregation and it is given to the apps as the input. We extract the evidences and then combine them for ranking fraud detection. Already done methods focus on learning a global ranking for the apps. Other methods depend on the training data and are difficult to be identified [5]. An app has a particular leading session in which the app is at the peak of its lifecycle. This has three phases namely the rising phase, maintaining phase and the recession phase. A particular app rises to a peak period, maintains that position and then decreases till the end of the session.

DOI: 10.15680/IJIRCCE.2015. 0401001

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Vol. 4, Issue 4, April 2016

IV.HUMMINGBIRD ALGORITHM

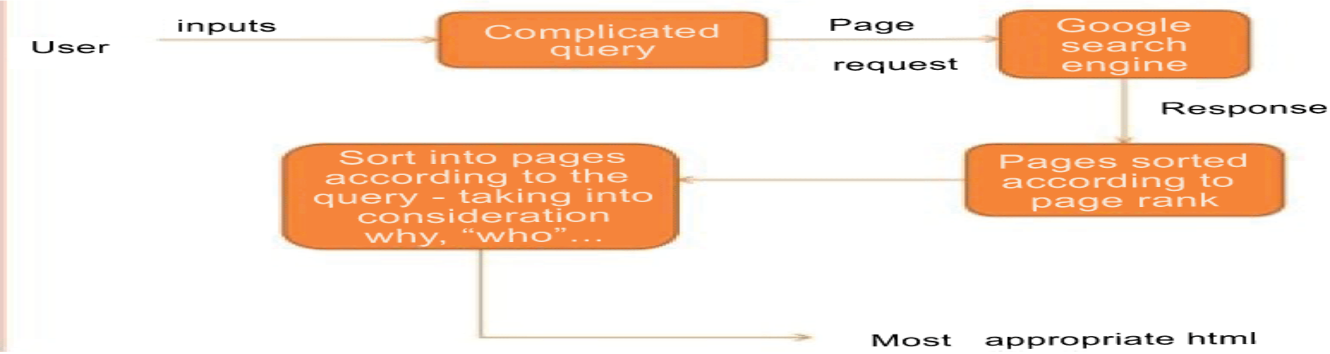
User
inputs
Complicated query
Page
request
Google search engine
Response
Pages sorted according to page rank
Sort into pages according to the query - taking into consideration why, "who"…
Most appropriate html

Figure 2: A working model of Google hummingbird algorithm.

Figure 1 represents the model of Hummingbird algorithm. It has the ability to quickly analyses longer and more complex questions and provides the best answer to the searches with the fewest possible clicks. It appears to be an update that underlying engine along the lines of Google's increased ability to map synonyms over time. It better focuses on the meaning behind the words. It provides a more human way to interact with users and provide a more direct answer unlike its previous versions like Panda and Penguin.[6]. It uses user information like previous download, geographical location etc. The user inputs the query in to the page and a request is sent to the search engine. After the response is received the pages are sorted according to their particular ranking and they are again sorted into pages according the query after taking into consideration "why, who, what, when" etc. The same search may yield different results to each user based on the circumstances. It can handle long complex search strings. It starts with answering the questions. This is called "conversational search". It uses Google's Knowledge Graph. A webpage's ranking is determined by analyzing the ranking of all other web pages in question. It basically uses intelligent ranking of web pages. It indexes faster and provides pages that are more recent. It relationally links search queries and web documents. It pays attention to each word in a query ensuring that the whole query-the whole conversation or meaning is taken into account not only the particular words [7].

Expanses the use of Knowledge graph so that the Google answers more complex search queries and also improves the follow up search process. For Example, if we first search "Qutb Minar" and again search "How tall is it?" Google will understand the context of the second query. It also enables more comparison between various items in context. It also provided geo-location enhancement which provides an answer to our query to our nearest location. Voice search and Android phone voice synchronization and improved and are likely to improve in the near years.

V. SESSION TRACKING ALGORITHM

This session tracking concept [8] is used in the proposed system to identify the users that are trying to misuse the particular App. There are three typical solutions to this problem: cookies, URL rewriting, and hidden form fields. You can use cookies to store an ID for a downloading session; with each subsequent connection, you can look up the current session ID and then use that ID to extract information about that session from a lookup table on the server machine. URL rewriting is a moderately good solution for session tracking and even has the advantage that it works when browsers don't support cookies or when the user has disabled them. The users that are using the App and downloading it are provided with a session each and they are continuously been tracked by the admin with the help of a session tracking algorithm. A cookie is assigned to each user as a session starts and it is been tracked as the user is continuously using the App. When a number of users are using the system by downloading and uploading the Apps, even when a particular user is found to be misusing the Apps among all other users, he is blocked with the help of a session tracking algorithm and all the user details are sent to the Admin immediately and the user is blocked from accessing the apps. The user is notified of the block and is permitted to access other apps. The number of times or the hits a particular user is using the App is being recorded with which the overall misusing of the App is calculated.

DOI: 10.15680/IJIRCCE.2015. 0401001

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Vol. 4, Issue 4, April 2016

VI. MODULE IMPLEMENTATION

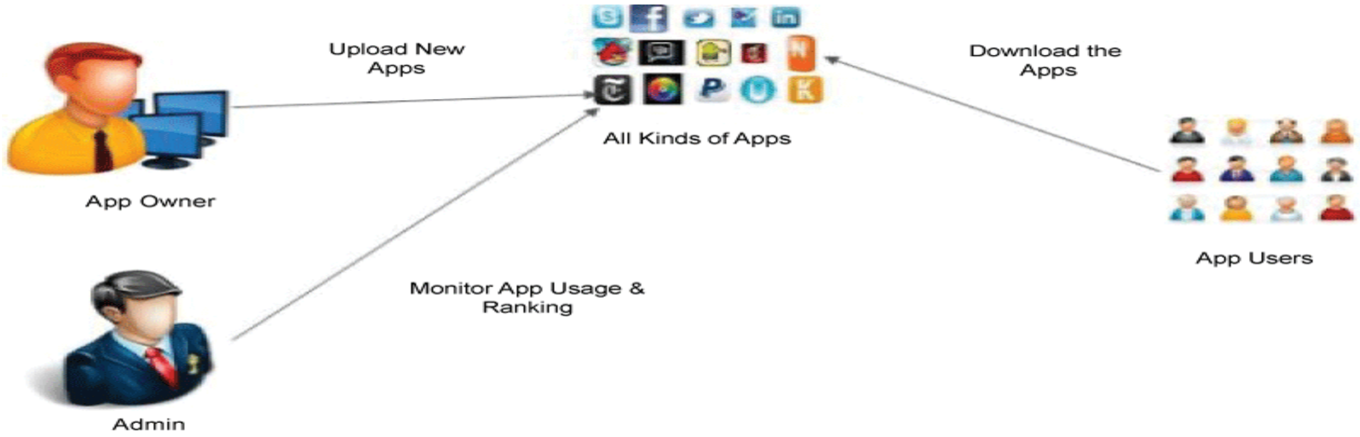

Figure 3: Architecture diagram.

The modules in the system are App owner, App Admin, App user. The App user first requires a username and password to download the App. If the user is new to the system the registration is done and username and password are sent to the Admin. After the login details are received, the user can login and search for the Apps under a specified platform in the home page. The user is given a option to choose a mobile type for which the Apps are listed in the page. When the user is ready to download the secret key is requested to the Admin for verification. The secret key is in a waiting state till the Admin approves the secret key. After the secret key is received, the user can download the app with the help of a secret key. The App owner also requires a login to upload the App. The app owner can also register with the page. After registering the owner uploads the App by providing the description regarding the particular App. The owner also can monitor the Apps that are leading in the current leader board. The owner can also view the fraud details with the time and the details of the user that is found to misuse the App.

The Admin maintains the overall management of the Apps that are uploaded by the various App owners. The Admin also has a login to view all the details regarding the various apps and its details. The Admin views the various requests for the secret key by the App users and assigns a secret key to each user on a specific range. The secret key is generated using a random number algorithm [9]. And is sent to the user who with the help of this secret key downloads the particular App and uses it. After the user has downloaded the App the ranking of the app increases. The user is provided with a limitation of a specific number of times to download the App. Each and every time the App is downloaded the number of times decreases correspondingly. After the constraint is hit the App user is blocked and is showed a message stating that the user has been blocked for misusing the app. The user can download other similar Apps. The evidence for fraud also can be viewed by the Admin. The App name, description and the Fraud IP address are displayed to the Admin to block the particular user. The date and time at which the activity was performed is saved for further verification. All the owners of the App are also available in the Admin login. The Admin can view the various users that are using the Apps available. The hits that a specific App receives are depicted in a graph that is available for viewing only by the Admin.

Minimizing total transmission energy ii) maximizing network lifetime. The first metric focuses on the total transmission energy used to send the packets from source to destination by selecting the large number of hops criteria. Second metric focuses on the residual batter energy level of entire network or individual battery energy of a node [1].

VI. CONCLUSION

In this paper, we developed a fraud detection system for mobile Apps. We showed that fault ranking can occur by spam reviews and app recommendations and the fault ranking can prove to be a major issue as it involves placing the app in a false position in the leader board. Thus, we identified the evidences and performed an aggregation with the type of evidences. The Humming bird algorithm was implemented in performing the search of the apps and viewing its relevant details. The ranking was increased with the help of this particular algorithm and it prevented the user from downloading the apps multiple number of times. The session tracking algorithm identifies the session that are currently in use and helps to block the current user. Experimental results showed the effectiveness of the proposed approach. We also propose to find the relationship among various types of evidences that are being used to find out the rank detection.[10]

REFERENCES

1. L Azzopardi, M Girolami, et al. Investigating the relationship between language model perplexity and IR precision-recall measures, in Proc. 26th Int. Conf. Res. Develop. Inform. Retrieval, 2003; 369–370.
2. L Barbosa, J Freire, Siphoning Hidden-Web Data through Keyword-Based Interfaces. In Proc. of SBBD, 2004; 309–321.
3. Z Hengshu, X Hui, et al. Discovery of ranking fraud for mobile apps. IEEE Transactions on knowledge and data engineering, 2014.
4. Rahman, S Huang, HV.Faloutsos. Detecting malicious Facebook applications. IEEE transactions on networking volume, 2015.
5. Z Hengshu, X Hui Xiong, et al. Discovery of ranking fraud for mobile apps. IEEE Transactions on knowledge and data engineering, 2014.
6. http://www.steamfeed.com/google-hummingbird-mean-future-seo/
7. http://www.slideshare.net/PriyodarshiniDhar/google-hummingbird-algorithm-ppt
8. http://www.tutorialspoint.com/servlets/servlets-session-tracking.html.
9. https://en.wikipedia.org/wiki/Random_number_generation
10. H Zhu.H.xiong, et al. Ranking fraud detection for mobile Apps: A holistic view," in Proc. 22nd ACM Int. Conf. Inform. Knowl. Manage. 2013; 619-628.

DOI: 10.15680/IJIRCCE.2015. 0401001