Ranking Fraud Detection and prevention On Relationship Among Rating Review & Ranking

Neha Hete[1], Saurabh A.Chepe[2]
e

[1] *Miss. Neha S. Hete, Maharashtra, India*
[2] *Mr. Saurabh A. Chepe Maharashtra, India,*

ABSTRACT

Ranking fraud in the mobile App market refers to fraudulent or deceptive activities which have a purpose of bumping up the Apps in the popularity list. Indeed, it becomes more and more frequent for App developers to use shady means, such as inflating their Apps' sales or posting phony App ratings, to commit ranking fraud. While the importance of preventing ranking fraud has been widely recognized, there is limited understanding and research in this area. To this end, in this paper, we provide a holistic view of ranking fraud and propose a ranking fraud detection system for mobile Apps. Specifically, we first propose to accurately locate the ranking fraud by mining the active periods, namely leading sessions, of mobile Apps. Such leading sessions can be leveraged for detecting the ***local*** *anomaly instead of global anomaly of App rankings. Furthermore, we investigate three types of evidences, i.e., ranking based evidences, rating based evidences and review based evidences, by modeling Apps' ranking, rating and review behaviors through statistical hypotheses tests. In addition, we propose an optimization based aggregation method to integrate all the evidences for fraud detection.The mobile app recommendation for Finally, we evaluate the proposed system with real-world App data collected from the iOS App Store for a long time period. In the experiments, we validate the effectiveness of the proposed system, and show the scalability of the detection algorithm as well as some regularity of ranking fraud activities.*

Keywords: *Mobile Apps, Ranking Fraud Detection, Evidence Aggregation, Historical Ranking Records, Rating and Review , Recommendate apps*

1. INTRODUCTION

The number of mobile Apps has grown at a breath taking ate over the past few years. For example, as of the end of April 2013, there are more than 1.6 million Apps at Apple's App store and Google Play. To stimulate the development of mobile Apps, many App stores launched daily App leader boards, which demonstrate the chart rankings of most popular Apps. Indeed, the App leader board is one of the most important ways for promoting mobile Apps. A higher rank on the leader board usually leads to a huge number of downloads and million dollars in revenue. Therefore, App developers tend to explore various ways such as advertising campaigns to promote their Apps in order to have their Apps ranked as high as possible in such App leader boards. However, as a recent trend, instead of relying on traditional marketing solutions, shady App developers resort to some fraudulent means to deliberately boost their Apps and eventually manipulate the chart rankings on an App store. This is usually implemented by using so-called "bot farms" or "human water armies" to inflate the App downloads, ratings and reviews in a very short time. For example, an article from Venture Beat reported that, when an App was promoted with the help of ranking manipulation, it could be propelled from number 1,800 to the top 25 in Apple's top free leader board and more than 50,000-100,000 new users could be acquired within a couple of days. In fact, such ranking fraud raises great concerns to the mobile App industry. For example, Apple has warned of cracking down on App developers who commit ranking fraud in the Apple's App store.

Ranking fraud in the mobile App market refers to fraudulent or deceptive activities which have a purpose of bumping up Apps in the popularity list. Indeed, it becomes more and more frequent for App developers to use shady means, such as inflating their Apps' sales or posting phony App ratings, to commit ranking fraud. While the importance of preventing ranking fraud has been widely recognized, there is limited understanding and research in this area. To this end, in this paper, we provide a holistic view of

ranking fraud and propose a ranking fraud detection system for mobile Apps. Specifically, we first propose to accurately locate the ranking fraud by mining the active periods, namely leading sessions, of mobile Apps. Such leading sessions can be leveraged for detecting the local anomaly instead of *global* anomaly of App rankings. Furthermore, we investigate three types of evidences, i.e., ranking based evidences, rating based evidences and review based evidences, by modeling Apps' ranking, rating and review behaviors through statistical hypotheses tests. In addition, we propose an optimization based aggregation method to integrate all the evidences for fraud detection. Finally, we evaluate the proposed system with real-world App data collected from the iOS App Store for a long time period. In the experiments, we validate the effectiveness of the proposed system, and show the scalability of the detection algorithm as well as some regularity of ranking fraud activities

.In this paper ,we present client server architecture ,where its client collect the application usage records and periodically uploads them to the server .The user can use the client to browse and install the application recommended for her .To preserve the users privacy the device ID was used to identify the application user.

2. LITERATURE REVIEW

In this paper, a ranking fraud detection system for mobile Apps. Specifically, first showed that ranking fraud happened in leading sessions and provided a method for mining leading sessions for each App from its historical ranking records. Then, identified ranking based evidences, rating based evidences and review based evidences for detecting ranking fraud. Moreover, proposed an optimization based aggregation method to integrate all the evidences for evaluating the credibility of leading sessions from mobile Apps. An unique perspective of this approach is that all the evidences can be modeled by statistical hypothesis tests, thus it is easy to be extended with other evidences from domain knowledge to detect ranking fraud. Finally, validate the proposed system with extensive experiments on real-world App data collected from the Apple‟s App store[1]

The authors present AppJoy, a system that makes personalized mobile application recommendations. The novel feature of AppJoy is that it measures how the applications are actually used, and the usage scores are then used by a Collaborative Filter (CF) algorithm to make personalized recommendations. This is analogous to the "vote by your feet" approach, in which what the user does matters more when profiling her application needs. Compared with other solutions, AppJoy is completely automatic without requiring manual input and AppJoy is adaptive to the changes of the user‟s application taste. To the best of our knowledge, AppJoy is the first mobile application discovery system that leverages the user‟s actual application usage patterns. AppJoy employs a client-server architecture, where its client collects the application‟s usage records and periodically uploads them to the server. The AppJoy server runs the CF algorithm that calculates recommendations for all users on a daily basis. The user can use the AppJoy client to browse and install the applications recommended for her .To preserve the user‟s privacy, only the device ID was used to identify the application users..[4]

With the rapid prevalence of smart mobile devices, the number of mobile Apps available has exploded over the past few years. To facilitate the choice of mobile Apps, existing mobile App recommender systems typically recommend popular mobile Apps to mobile users. However, mobile Apps are highly varied and often poorly understood, particularly for their activities and functions related to privacy and security. Therefore, more and more mobile users are reluctant to adopt mobile Apps due to the risk of privacy invasion and other security concerns. To fill this crucial void, in this paper, to develop a mobile App recommender system with privacy and security awareness. The design goal is to equip the recommender system with the functionality which allows to automatically detect and evaluate the security risk of mobile Apps. Then, the recommender system can provide App recommendations by considering both the Apps‟ popularity and the users‟ security preferences. Specifically, a mobile App can lead to security risk because insecure data access permissions have been implemented in this App. Therefore, develop the techniques to automatically detect the potential security risk for each mobile App by exploiting the requested permissions. Then, the author propose a flexible approach based on modern portfolio theory for recommending Apps by striking a balance between the Apps‟ popularity and the users‟ security concerns, and build an App hash tree to efficiently recommend Apps[5.]

This paper aims to detect users generating spam reviews or review spammers. Identify several characteristic behaviors of review spammers and model these behaviors so as to detect the spammers. In particular, we seek to model the following behaviors. First, spammers may target specific products or product groups in order to maximize their imp- pact. Second, they tend to deviate from the other reviewers in their ratings of products. The author propose scoring methods to measure the degree of spam for each reviewer and apply them on an Amazon review dataset. then select a sub- set of highly suspicious reviewers for further scrutiny by our user evaluators with the help of a web based spammer evaluation software specially developed for user evaluation experiments. Our results show that our proposed ranking and supervised methods are effective in discovering spammers and outperform other baseline method based on helpfulness votes alone. finally show that the detected spammers have more significant impact on ratings compared with the unhelpful reviewers[9].

Ranking fraud in the mobile App business suggest to false or tricky exercises which have a motivation behind, knocking up the Apps in the fame list. Now a days, many shady means are used more frequently by app developers, such expanding their Apps' business or posting imposter App evaluations, to confer positioning misrepresentation. There is a limited understanding and research area for preventing ranking fraud. This paper gives a whole perspective of positioning misrepresentation and describes a Ranking fraud identification framework for mobile Apps. This work is grouped into three category. First is web ranking spam detection, second is online review spam detection and last one is mobile app recommendation. The Web ranking spam refers to any deliberate actions which bring to selected Web pages an unjustifiable favorable relevance or importance. Review spam is designed to give unfair view of some products so as to influence the consumers' perception of the products by directly or indirectly inflating or damaging the product's reputation[[15]**.**

3. PROPOSED TECHNIQUE

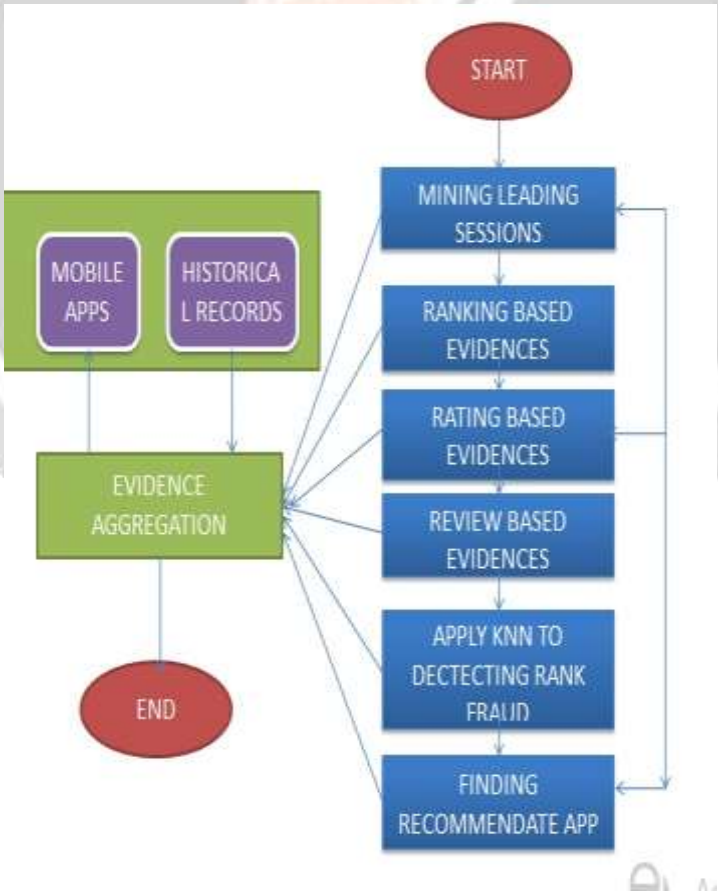

Fig1: The framework for fraud detection

Mining leading sessions

From the Apps historical rating , discovery of leading events is done which appeared for constructing leading sessions.

Ranking based evidence

By analysis of basic behavior of leading events for finding fraud evidences and also for the app historical ranking records.

Rating based evidence

As we know that rating is high in leaderboard considerably that is attracted by most of the mobile app users.

The rating during the leading sessions give rise to the anomaly patterns which happens during rating fraud.

Evidence agregation

At the admin side the evidence aggregation is calculated. The admin can easily know how many users are there for an app. And in the system the hit rate gets changed even when the user views the app.

Admin side implementation of the system:
Admin maintains the storage space status like app rating, approved users and sending secret key to the users. Every time a new registers in the store by giving his/ her details, the admin allots a secret key for that user. The user can login into his/her account using the secret key alone. this secret key is unique for each other and is generated. The admin will receive the user details from the fake ranking blocker and will be able to maintain a unique user list for particular app.thus the admin can provide the genuine app details such as rating and ranking by knowing the number of unique users for an app.
User side implementation of the system:
User can download apps for android, windows and various other kinds of mobile phones. when the user searches for a particular app, the searching is done and the most app and high ranked app is shown in the result page. he can view details about an app and download it.to prevent the fraud ranking its defined the user can rate the app only for five times. when he tries to rate the app for the sixth time our internal architecture will block the users download and sends the users system configuration details and the details of the app he is trying to download to the admin. This function is carried out by the fake rank blocker in the system. now admin can easily know how many users are there for an app . Also in the existing system the hit rate gets changed even when the user views an app but this system makes sure that hate rate is affected only when an app is downloaded .The fraud detection is done by K means algorithm.

K-means algorithm:

The most common algorithm uses an iterative refinement technique. Due to its ubiquity it is often called the ***k*-means algorithm**; it is also referred to as **Lloyd's algorithm**, particularly in the computer science community.

Given an initial set of *k* means $m_1^{(1)},\ldots,m_k^{(1)}$ (see below), the algorithm proceeds by alternating between two steps

Assignment step: Assign each observation to the cluster whose mean yields the least within-cluster sum of squares (WCSS). Since the sum of squares is the squaredEuclidean distance, this is intuitively the "nearest" mean.[8] (Mathematically, this means partitioning the observations according to the Voronoi diagram generated by the means).

$$S_i^{(t)} = \{x_p : \|x_p - m_i^{(t)}\|^2 \leq \|x_p - m_j^{(t)}\|^2 \ \forall j, 1 \leq j \leq k\},$$

where each x_p is assigned to exactly one $S^{(t)}$, even if it could be assigned to two or more of them.

Update step: Calculate the new means to be the centroids of the observations in the new clusters.

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

Since the arithmetic mean is a least-squares estimator, this also minimizes the within-cluster sum of squares (WCSS) objective.

The algorithm has converged when the assignments no longer change. Since both steps optimize the WCSS objective, and there only exists a finite number of such partitionings, the algorithm must converge to a (local) optimum. There is no guarantee that the global optimum is found using this algorithm.

The algorithm is often presented as assigning objects to the nearest cluster by distance. The standard algorithm aims at minimizing the WCSS objective, and thus assigns by "least sum of squares", which is exactly equivalent to assigning by the smallest Euclidean distance. Using a different distance function other than (squared) Euclidean distance may stop the algorithm from converging.[citation needed] Various modifications of k-means such as spherical k-means and k-medoids have been proposed to allow using other distance measures.

Further discussion about the proposed approach

The proposed system overcomes the disadvantages in the existing system by having the following advantages. The first advantage is that the user will be able to increase an app hit rate by downloading the app alone.so there is possibility for the app developer to make a particular user to download their app in order to increase the hit rate of the app.so the system restricts the maximum number of times a user can download to app to five .If the user attempts to download the app for sixth time the user is suspected to have illegally increase the hit rate of the app. The user details and his/her system configuration are sent to the admin along with the app details .The users are allowed to login to their account using the secret key allotted to them by the admin .The admin can maintain a unique list of unique users and knows exact the number of users for an app .When a user account is suspected for the practice of fraud activity ,the system configuration details of that user are sent to the admin thus the user can be tracked by the admin .The ranking details provided by the admin can guide the user to download a genuine app.

1.Registration:

If user want to join the store then he can register with his information.

Fig2: registration

2.Login**:**After the successfully registration the user can login.

Fig3: login

3.Leading Sessions:

In leading sessions ,the various application and games are available for the user. He can see the details of every app which are available in store. The leading sessions of mobile app signify the period of popularity. the issue of identifying ranking fraud is to identify deceptive leading sessions.

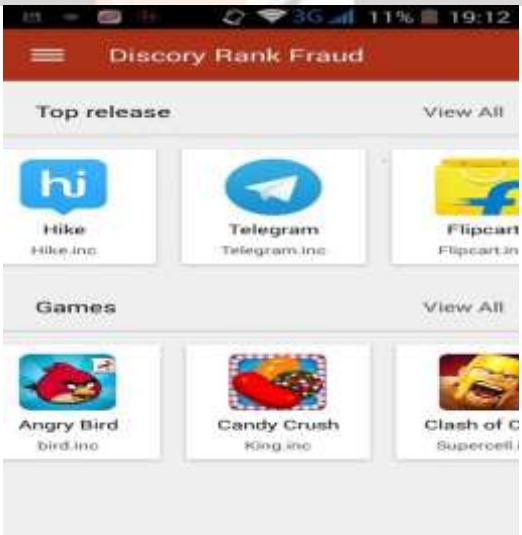

Fig4: leading sessions

4.Identify the leading sessions:

From the Apps historical rating , discovery of leading events is done which appeared for constructing leading sessions .In leading sessions ,the top k apps are available.

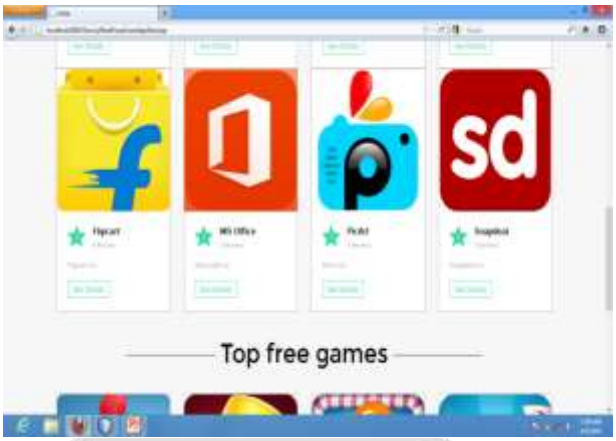

Fig5: identify the leading sessions

5.Rating based Evidence:

As we know that rating is high in leaderboard considerably that is attracted by most of the mobile app users .The rating during the leading sessions give rise to the anomaly patterns which happens during rating fraud. The rating is given to the app in which way

Fig 6 :Rating based evidence

But the user can rate the app only for five times. when he tries to rate the app for the sixth time our internal architecture will block the users.

You are blocked for Missusing the Application and you are Fraud know....!

Fig 7: user is blocked

6.Evidence Agreegation**:**

At the admin side the evidence aggregation is calculated. The admin can easily know how many users are there for an app. And in the system the hit rate gets changed even when the user views the app.

Result

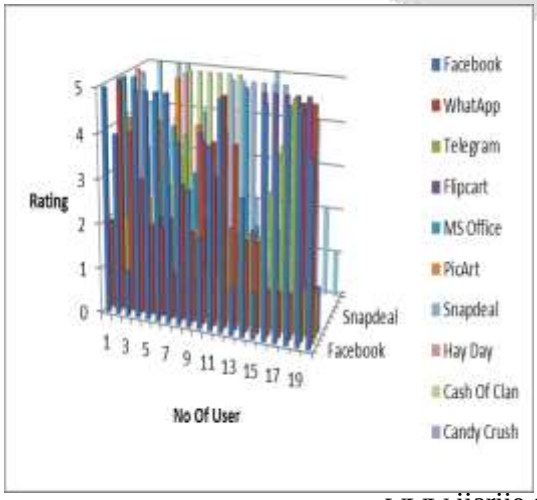

Fig8 : The distribution apps w.r.t different number of ratings.

The above figure shows the distribution of the number of app with respect to different ratings in these data sets. In this ,we can see that the distribution of app rating is not even, which indicates that only a small percentage of apps are very popular.

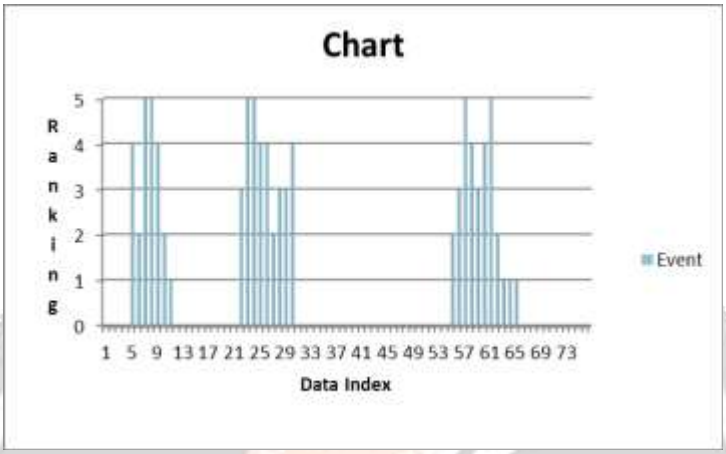

Fig 9 :The distribution of the number of apps w.r.t. different numbers of leading events

In fig 9 the distribution of the number of apps with respect to different number of leading events. Event means when the user loin into the system and he downloaded or installed or give rating the apps. The figure shows the apps records i. e 1 to 73 when user can login the system and he downloaded or rated or installed any app. In this figure , from 5 to 12 records, the event is created with high ranking to the apps. but 13 to 20 no event is created. Same as 21 to 29 the and 54 to 64 event is created.

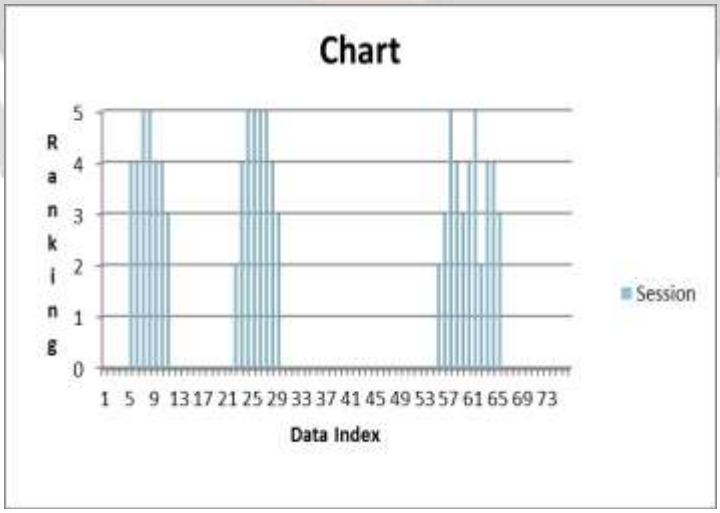

Fig 10 : The distribution of the number of apps w.r.t. different numbers of leading sessions

The fig 10 shows the distribution of the number of apps with respect to different numbers of leading sessions. Leading sessions means the user only visit the app not perform any action. In above figure from 5 to 11 the users visit the apps but from 12 to 20 users dont visit the apps.

Comparison

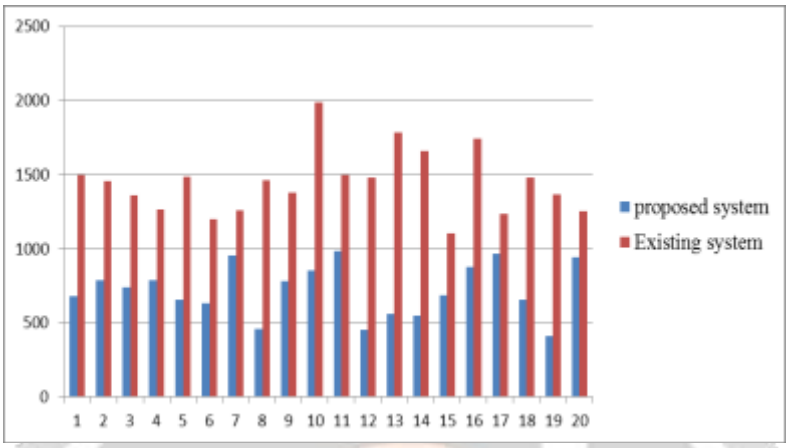

Fig 11: Time comparison between existing and proposed system

In fig the time required for detecting ranking fraud is given. In proposed system the less time is required because the K-means algorithm is used but in existing system more time is required without k-means algorithm. Time taken in ms.

4. CONCLUSIONS

In this paper, we developed and prevented a ranking fraud detection system for mobile Apps. Specifically, we first showed that ranking fraud happened in leading sessions and provided a method for mining leading sessions for each App from its historical ranking records. Then, we identified ranking based evidences, rating based evidences and review based evidences for detecting ranking fraud. It also provides a way to track the user who involved in ranking fraud and makes the admin who know the exact number of users for an app.

5. REFERENCES

[1]. Discovery of Ranking fraud for mobile apps. Hengshu Zhu,Hui Xiong,Senior members,IEEE,Yong Ge,and Enhong Chen,Senior member,IEEE,IEEE transactions on knowledge and data engineering,vol .27,No.1,January 2015.

[2]. N. Spirin and J. Han. "Survey on web spam detection: principles and algorithms" SIGKDD Explor. Newsl., 13(2):50–64, May 2012.

[3]. 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.

[4]. B. Yan and G. Chen. "Appjoy: personalized mobile application discovery" In Proceedings of the 9th international conference on Mobile systems, applications, and services, MobiSys '11, pages 113–126, 2011.

[5]. S.K.Ram Kumar#1, Dr. N. Lakshmi Narasimman*2. "Security Awareness of Mobile Application for Discovering Fraud Rank" In proceeding S.K.Ram Kumar et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 7 (3) , 2016, 1595-1599A.

[6]. G.Subramani , R.Selvarasan , "Hierarchy fraud detection for mobile apps and efficient searching apps" proceeding international journal of applied theoretical and technology volume 1,issue 3,pp,45-48 ,august 2015.

[7]. Y. Ge, H. Xiong, C. Liu, and Z.-H. Zhou. "A taxi driving fraud detection system" In Proceedings of the 2011 IEEE 11th International Conference on Data Mining, ICDM '11, pages 181–190, 2011.

[8]. 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.

[9]. 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.

[10]. A. Mukherjee, A. Kumar, B. Liu, J. Wang, M. Hsu, M. Castellanos, and R. Ghosh. "Spotting opinion spammers using behavioral footprints" In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '13, 2013.

[11]. 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.

[12]. M. N. Volkovs and R. S. Zemel. "A flexible generative modelfor preference aggregation" In Proceedings of the 21st international conference on World Wide Web, WWW '12, pages 479–488, 2012.

[13] Z.Wu, J.Wu, J. Cao, and D. Tao. "Hysad: a semi-supervised hybrid shilling attack detector for trustworthy product recommendation" In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '12, pages 985–993, 2012.

[14]. S. Xie, G. Wang, S. Lin, and P. S. Yu. "Review spam detection via temporal pattern discovery" In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '12, pages 823–831, 2012.

[15]. H. Zhu, H. Cao, E. Chen, H. Xiong, and J. Tian. "Exploiting enriched contextual information for mobile app classification" In Proceedings of the 21st ACM international conference on Information and knowledge management, CIKM '12, pages 1617–1621, 2012.

[16]. Klementiev, D. Roth, and K. Small. "Unsupervised rank aggregation with distance-based models" In Proceedings of the 25th international conference on Machine learning, ICML '08, pages 472– 479, 2008.
[17]. A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. "Detecting spam web pages through content analysis" In Proceedings of the 15th international conference on World Wide Web, WWW '06, pages 83–92, 2006.

[18]. T. L. Griffiths and M. Steyvers. "Finding scientific topics" In Proc.nof National Academy of Science of the USA, pages 5228–5235, 2004.

[19]. L. Azzopardi, M. Girolami, and K. V. Risjbergen. "Investigating the relationship between language model perplexity and ir precision recall measures" In Proceedings of the 26th International Conference on Research and Development in Information Retrieval (SIGIR'03), pages 369–370, 2003.