Janani.S *et al*, International Journal of Computer Science and Mobile Applications,
Vol.5 Issue. 10, October- 2017, pg. 181-187 **ISSN: 2321-8363**
Impact Factor: 4.123

FRAUD DETECTION USING REDDIT RANKING ALGORITHM

Janani.S, Dr. R.Manickachezian

Research Scholar, Dr. Mahalingam Centre for Research and Development, NGM College, Pollachi,
Department of Computer Science, N G M College (Autonomous), Pollachi, Coimbatore - 642001,
Email: jananiakshya@gmail.com, chezian_r@yahoo.co.in

ABSTRACT

Traditional methods of data analysis have long been used to detect fake reviews. Early data analysis techniques were oriented toward extracting quantitative and statistical data characteristics. Some of these techniques facilitate useful data interpretations and can help to get better insights into the process behind data. To go beyond a traditional system, a data analysis system has to be equipped with considerable amount of background data, and be able to perform reasoning tasks involving that data. In effort to meet this goal researchers have turned to the fields of machine learning and artificial intelligence. A review can be classified as either fake or genuine either by using supervised and/or unsupervised learning techniques. These methods seek reviewer's profile, review data and activity of the reviewer on the Internet mostly using cookies by generating user profiles. Using either supervised or unsupervised method gives us only an indication of fraud probability. In the proposed system here we are introducing a cross combined technology for Reddit Ranking Algorithm which comes under opinion mining category. Here we use Advanced Text categorization (ATC) with artificial neural network (ANN). We propose a deep data analysis model to identify fake users. This can be identified by using Ranking Process. Each and every comment will be categorized in this category. After the categorization, used count will be taken from the number of comments. In case of the user comments are around the range the user's rating will be accepted. Else the user will be blacklisted and the user's review will be removed from the rating list. This makes the system work perfectly with fine actual reviews.

Keywords: Machine learning, Artificial intelligence, Advance Text categorization, user comments

1. Introduction

To recognize a few trade mark practices of audit spammers and model these practices to distinguish the spammers. Specifically, we try to display the accompanying practices. To start with, spammers may target particular items or item bunches keeping in mind the end goal to amplify their effect. Second, they tend to go astray from alternate commentators in their appraisals of items. We propose scoring techniques to quantify the level of spam for every analyst and apply them on an Amazon audit dataset. We at that point select a subset of exceptionally suspicious commentators for assist examination by our client evaluators with the assistance of an electronic spammer assessment programming uniquely created for client assessment tests. Our outcomes

demonstrate that our proposed positioning and regulated techniques are powerful in finding spammers and outflank other gauge strategy in light of accommodation votes alone. We at long last demonstrate that the identified spammers have more huge effect on evaluations contrasted and the unhelpful analysts. Distinguishing audit spam is a testing errand as nobody knows precisely the measure of spam in presence. Because of the transparency of item audit destinations, spammers can act like diverse clients (known as "sock puppeting") contributing spammed surveys making them harder to destroy totally. Spam surveys typically look flawlessly ordinary until one contrasts them and different audits of similar items to distinguish survey remarks not predictable with the last mentioned. The endeavours of extra correlations by the clients influence the discovery to undertaking repetitive and non-minor. One approach taken by audit site, for example, Amazon.com is to enable clients to mark or vote the surveys as supportive or not. Tragically, this still requests client endeavours and is liable to mishandle by spammers. The best in class way to deal with audit spam identification is to regard the surveys as the objective of recognition. This approach speaks to an audit by survey, analyst and item level highlights, and prepares a classifier to recognize spam surveys from non-spam ones.

2. Related Works

The business achievement of Android application markets, for example, Google Play [1] and the motivation display they offer to well known applications, make them engaging focuses for deceitful and pernicious practices. Some fake engineers misleadingly help the hunt rank and prominence of their applications (e.g., through phony surveys and counterfeit establishment tallies) [2], while vindictive designers utilize application advertises as a platform for their malware [3]– [6]. The inspiration for such practices is affect: application fame surges convert into money related advantages and sped up malware expansion. Fake designers oftentimes abuse swarm sourcing destinations (e.g., Freelancer [7], Fiverr [8], Best App Promotion [9]) to contract groups of willing laborers to confer misrepresentation all in all, copying sensible, unconstrained exercises from random individuals (i.e., "swarm turfing" [10]). We call this conduct "look rank extortion". Moreover, the endeavors of Android markets to recognize and expel malware are not generally effective. For example, Google Play utilizes the Bouncer framework [11] to expel malware. Be that as it may, out of the 7, 756 Google Play applications we broke down utilizing VirusTotal [12], 12% (948) were hailed by no less than one hostile to infection apparatus and 2% (150) were distinguished as malware by no less than 10 instruments. Past portable malware identification work has concentrated on powerful investigation of application executables [13]– [15] and in addition static examination of code and consents [16]– [18]. In any case, late Android malware examination uncovered that malware develops rapidly to sidestep hostile to infection instruments [18].

3. Existing System

In existing analysis, the data are assumed to be sampled from a mixture distribution with K components corresponding to the K spammers to be recovered. Let (X1, ..., Xp) denote a random 1 ×p vector of explanatory variables or features, and let Y ∈ {1, ..., K} denote the unknown component or spammer label. Given a sample of X values, the goal is to estimate the number of spammers K and to estimate, for each observation, its spammer label Y. Suppose we have data X = (xij) on p explanatory variables (for example, genes) for n observations (for example, tumor m RNA samples), where xij denotes the realization of variable Xj for observation i and xi = (xi1,...,.xip) denotes the data vector for observation i, i = 1,...,n, j = 1,...,p. We consider spammer procedures that partition the learning set $\mathcal{L}$ = {x1,...,xn} into K spammers of observations that are 'similar' to each other, where K is a user-prespecified integer. More specifically, the spammering $\mathcal{P}(\cdot;\mathcal{L})$ assigns class labels $\mathcal{P}$ (Xi; $\mathcal{L}$) $= \hat{y}_i$ to each observation, where $\hat{y}_i \in \{1,\ldots,K\}$. Spammering procedures generally operate on a matrix of pair wise dissimilarities (or similarities) between the observations to be spammered, such as the Euclidean or Manhattan distance matrices. A partitioning of the learning set can be produced directly by partitioning spammering methods (for example, k-means, partitioning around medoid (PAM), self-organizing maps (SOM)) or by hierarchical spammering methods, by 'cutting' the dendrogram to obtain K 'branches' or spammers. Important issues, which will only be addressed briefly in this article, include: the selection of observational units, the selection of variables for defining the groupings, the transformation and standardization of variables, the choice of a similarity or dissimilarity measure, and the choice of a spammering method. Our main concern here is to estimate the number of spammers K.

It can find out only the k-means spammering aims to partition **n** observations into **k** spammers in which each observation belongs to the spammer with the nearest mean, serving as a prototype of the spammer. This results in a partitioning of the data space into less result.

k- Number of data

n – Repeated comments by a user for a same product

4. Proposed Architecture – Reddit Ranking Algorithm

Let total comment be TC, Considering the posted comment from ANN will be initialized as PC as positive comment and NC as the negative comment. As per Opinion Mining 6 will be the least consideration and here the consideration will be 9 parameters. Buffer and Array will be FD[num]arr. As per commitment FD[>=60]arr will fraudulent user

4.1 Algorithm steps

Step 1 : Counting total number of comments for a user during their post =TC;
Step 2: Calculating 1:8 Ration for Positive boost and 8:1 for negative boost;
Step 3: If 1<=8 & 1*8 = >8 means, the user may exceed with PC(Positive Comment) and 20 marks will be added in the fraudulent range, else No;
Step 4: If 8<=1 & 8*1=>1 means, the user may exceed with NC (Negative Comment) and 20 marks will be added in the fraudulent range, else No;
Step 5: Fraudulent Range will stored in an array namely FD[0]arr; check up to >60;
Step 6: If product purchased & PC >=5 FD[20]arr else 0, FD[0]arr;
Step 7: Also product purchased & NC>=5 FD[20]arr else 0,FD[0]arr
Step 8: check FD[num]arr >60
Step 9: Check not purchased PC >= 3 FD[20]arr if <3 FD[0]arr;
Step 10 : As per previous check not purchased NC>=3 FD[20]arr if<3 FD[0]arr;
Step 11: check FD[num]arr >60 : These above given 6 are the important criteria as per opinion mining concept in the mining technique.
Step 12: Date of creation be DC: Initially DC will be in 0. Each day DC = i=i++. So that DC =[i+1];
Step 13 : Go to step 3, Step 4, for Condition 1; FD[20]arr (r) FD[0]arr;
Step 14 : Go to step 6, Step 7 for Condition 2; FD[20]arr (r) FD[0]arr;
Step 15: Go to Step 9, Step 10 for Condition 3; FD[20]arr (r) FD[0]arr;
Step 16: Considering Step 13, Step 14, Step 15 : if any of 2 conditions marked as FD[20]arr (r) FD[0]arr;
Step 17: In case of true FD[20]arr else FD[0]arr;
Step 18: Number of login = 0; if username and password is valid = Number Login = i+1;
Step 19 : Go to step 3, Step 4, for Condition 1; FD[20]arr (r) FD[0]arr;
Step 20: Go to step 6, Step 7 for Condition 2; FD[20]arr (r) FD[0]arr;
Step 21: Go to Step 9, Step 10 for Condition 3; FD[20]arr (r) FD[0]arr;
Step 22 : Go to 17; FD[20]arr else FD[0]arr;
Step 23 : In case of true FD[10]arr else FD[0]arr;
Step 24: Check number of same IP from DC >=3 : FD[10]arr else FD[0]arr;
Step 25: Calculating Fraudulent user:
Step 26 : IF FD[>=60]arr Fraudulent user: TC, PC, NC,DC will be cleared from the DB.
Step 27: Else No change in DB

5. Ranking Procedure

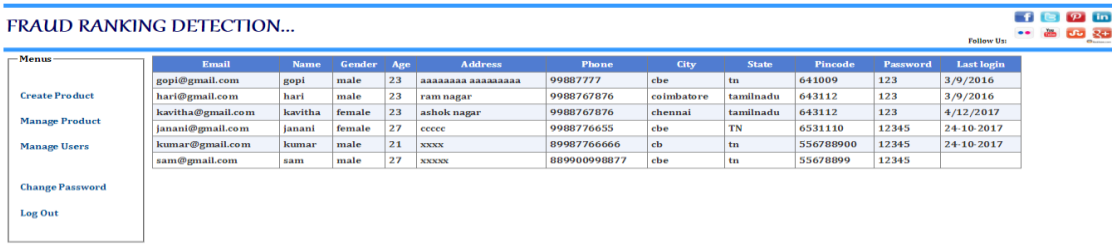
In the above formula the parameters are defined in a following way: p is the observed fraction of positive ratings. n is the total number of ratings. zα/2 is the (1-α/2) quantile of the standard normal distribution.

$$\frac{\hat{p} + \frac{1}{2n}z^2_{1-\alpha/2} \pm z_{1-\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n} + \frac{z^2_{1-\alpha/2}}{4n^2}}}{1 + \frac{1}{n}z^2_{1-\alpha/2}}$$

Let's summarize the above in a following manner: The confidence sort treats the vote count as a statistical sampling of a hypothetical full vote by everyone. The confidence sort gives a comment a provisional ranking that it is 85% sure it will get to. The more votes, the closer the 85% confidence score gets to the actual score. Wilson's interval has good properties for a small number of trials and/or an extreme probability

6. Result And Discussion

The below given result shows the user using the above given method. fig 1.1



Email	Name	Gender	Age	Address	Phone	City	State	Pincode	Password	Last login
gopi@gmail.com	gopi	male	23	aaaaaaaa aaaaaaaaa	99887777	cbe	tn	641009	123	3/9/2016
hari@gmail.com	hari	male	23	ram nagar	9988767876	coimbatore	tamilnadu	643112	123	3/9/2016
kavitha@gmail.com	kavitha	female	23	ashok nagar	9988767876	chennai	tamilnadu	643112	123	4/12/2017
janani@gmail.com	janani	female	27	ccccc	9988776655	cbe	TN	6531110	12345	24-10-2017
kumar@gmail.com	kumar	male	21	xxxx	89987766666	cb	tn	556788900	12345	24-10-2017
sam@gmail.com	sam	male	27	xxxxx	889900998877	cbe	tn	55678899	12345	

7. Future Work

Even the system is working well a according to the commitment, still we need some enhancement to make the system more efficient and better. Being ecommerce application has been used in this application, as the content management system, the system need to meet out the latest technology. As per the study the ecommerce application will work efficiently on cloud computing, our current system will be supporting on cloud computing. Our future work will be based on the Green Computing. Green computing overcomes all the drawbacks in the cloud computing. Our system supports client server architecture also.

Conclusion

In this paper a system for detecting the fraudulent reviews has been developed. Initially the application environment is created with an admin. This ecommerce application has a product and review based communication between the seller and the user. Here the data training is done using the Artificial Neural Networks in which the admin can update the positive and negative words. Based on this data training the text is categorized in to positive, negative and neutral comments. Here text categorization is implemented for sentimental data analysis. These methods will analysis the input data from the data set. Each sentence will be

analyzed with text categorization methods. So that positive words and negative words will be compared accordingly. The fake reviews are identified through analyzing various parameters such as numbers of repeated comments, number of comments for the same product and so on. Once the repeated comments are identified and it is removed from the comment list. Thus the system works perfect and remove fraudulent users

References

[1] Gang Wang, Christo Wilson, Xiaohan Zhao, Yibo Zhu, Manish Mohanlal, Haitao Zheng, and Ben Y. Zhao. 2012. Serf and Turf: Crowdturfing for Fun and Profit. In Proceedings of ACM WWW. ACM,

[2] Jon Oberheide and Charlie Miller2012, Dissecting the Android Bouncer. SummerCon2012, New York.

[3] VirusTotal - Free Online Virus, Malware and URL Scanner. https: //www.virustotal.com/, Last accessed on May 2015.

[4] Iker Burguera, Urko Zurutuza, and Simin Nadjm-Tehrani. Crowdroid ACM, 2011: Behavior-Based Malware Detection System for Android. In Proceedings of ACM SPSM, pages 15–26..

[5] Asaf Shabtai, Uri Kanonov, Yuval Elovici, Chanan Glezer, and Yael Weiss. Andromaly 2012. : a Behavioral Malware Detection Framework for Android Devices. Intelligent Information Systems, 38(1):161–190

[6] Michael Grace, Yajin Zhou, Qiang Zhang, Shihong Zou, and Xuxian Jiang. Riskranker: Scalable and Accurate Zero-day Android Malware Detection. In Proceedings of ACM MobiSys, 2012.

[7] Bhaskar Pratim Sarma, Ninghui Li, Chris Gates, Rahul Potharaju, Cristina Nita-Rotaru, and Ian Molloy, 2012. Android Permissions: a Perspective Combining Risks and Benefits. In Proceedings of ACM SACMAT.

[8] Hao Peng, Chris Gates, Bhaskar Sarma, Ninghui Li, Yuan Qi, Rahul Potharaju, Cristina Nita-Rotaru, and Ian Molloy, 2012. Using Probabilistic Generative Models for Ranking Risks of Android Apps. In Proceedings of ACM CCS.

[9] S.Y. Yerima, S. Sezer, and I. Muttik Sept 2014. Android Malware Detection Using Parallel Machine Learning Classifiers. In Proceedings of NGMAST.

[10] Yajin Zhou and Xuxian Jiang. Dissecting Android Malware 2012.: Characterization and Evolution. In Proceedings of the IEEE S&P, pages 95–109. IEEE.

[11] Fraud Detection in Social Networks. https://users.cs.fiu.edu/ ~carbunar/caspr.lab/socialfraud.html.

[12] Google I/O 2013 - Getting Discovered on Google Play. www. youtube.com/watch?v=5Od2SuL2igA.

[13] Justin Sahs and Latifur Khan , 2012. A Machine Learning Approach to Android Malware Detection. In Proceedings of EISIC.

[14] Borja Sanz, Igor Santos, Carlos Laorden, Xabier Ugarte-Pedrero, Pablo Garcia Bringas, and Gonzalo Alvarez. ´ Puma Springer, 2013: Permission usage to detect malware in android. In International Joint Conference CISIS12-ICEUTE´ 12-SOCO´ 12 Special Sessions, pages 289–298.

[15] Junting Ye and Leman Akoglu Springer, 2015. . Discovering opinion spammer groups by network footprints. In Machine Learning and Knowledge Discovery in Databases, pages 267–282.

[16] Leman Akoglu, Rishi Chandy, and Christos Faloutsos 2013. Opinion Fraud Detection in Online Reviews by Network Effects. In Proceedings of ICWSM,

[17] Android Market API 2011. . https://code.google.com/p/ android-market-api/,

[18] Etsuji Tomita, Akira Tanaka, and Haruhisa Takahashi , October 2006. The worstcase time complexity for generating all maximal cliques and computational experiments. Theor. Comput. Sci., 363(1):28–42.

BIOGRAPHIES

Janani.S received her B.Sc (Computer Science) from PSG College Of Arts and Science, coimbatore,India. She completed her Master of Computer Science (MSc) from Sri Ramakrishna College Of Arts and Science for Women , Coimbatore, India. She completed her MBA(HR) from School Of Distance Education, Bharathiar University. Currently, she is a Research Scholar at Department of Computer Science, NGM College, Pollchi, India. She participated and presented paper in a International Conference. Her area of interest includes Data mining, web content mining, Opinion mining.

Dr. R.ManickaChezian received his M.Sc., degree in Applied Science from P.S.G College of Technology, Coimbatore, India in 1987. He completed his M.S. degree in Software Systems from Birla Institute of Technology and Science, Pilani, Rajasthan, India and Ph.D degree in Computer Science from School of Computer Science and Engineering, Bharathiar University, Coimbatore, India. He served as a Faculty of Maths and Computer Applications at P.S.G College of Technology, Coimbatore from 1987 to 1989. Presently, he has been working as an Associate Professor of Computer Science in N G M College (Autonomous), Pollachi under Bharathiar University, Coimbatore, India since 1989. He has published one-fifty papers in international/national journal and conferences: He is a recipient of many awards like DeshaMithra Award and Best Paper Award. Recently he received the award "Best Computer Science Faculty of the Year 2015" from Association of Scientists, Developers and Faculties. His research focuses on Network Databases, Data Mining, Distributed Computing, Data Compression, Mobile Computing, Real Time Systems and Bio-Informatics.

