International Journal on Future Revolution in Computer Science & Communication Engineering
Volume: 4 Issue: 4
ISSN: 2454-4248
654 – 662

An Implementation to Detect Fraud App Using Fuzzy Logic

Neha M. Puram
M.Tech (CSE) Student,
Department of Comp Tech,
YCCE, Nagpur
nehampuram95@gmail.com

Dr. K. R. Singh
HOD, Department of Comp Tech,
YCCE, Nagpur
singhkavita19@yahoo.com

Abstract — Fraudulent behavior is most popular in app stores like Google play store, Apple's app store, *etc.* The popularity information in app stores, such as chart rankings, user ratings, and user reviews, provides an extraordinary chance to recognize user experiences with mobile apps. Many fraud app detection tools are available these days which extract evidences of reviews and ratings to detect the fake apps with different approaches. But most of the existing tools work on two groups *i.e.*, good and bad. So, we propose a system that works on more than two groups namely, very bad, bad, neutral, good and very good. Each group has been assigned a score which will improve the differentiation of reviews and ratings. For this the proposed system uses fuzzy logic algorithm. We have performed experimentation on 80 app ids taken from App-Review-Dataset, results show that proposed method is efficient in terms of accuracy and time required for retrieval.

Keywords — Mobile apps, ratings, reviews, fuzzy logic, classification, fraud detection.

*****

I. INTRODUCTION

Now-a-days with increase craze towards android mobiles the craze of mobile applications has also increased. According to the recent study, the number of application in Google play store, which is also known as Android Market grew from 1 million to 3.5 million [18] till December 2017. Whereas, the number of application in Apple's play store are 2.2 million [19] from 2008 July to 2017 January. The app developers try false mechanism so that the app developed by them should have high rank in the app leaderboard. App leaderboard is the one which shows the chart ranking of the popular apps, and it is a way through which one can promote their mobile apps. High rank of the app in leaderboard leads not only to the large number of downloads of that app but also million dollars in revenue. So, the developers of the app try various methods to promote their apps like advertising which helps them to have higher rank in the app leaderboard. However, instead of using ethical mechanism to promote their apps, the app developers try unethical means to promote their apps which manipulates the chart ranking of the app in the leaderboard and hence the app is ranked high in the leaderboard. This kind of unethical mechanism is generally carried out using "internet water army". Internet water army is a group of internet ghostwriters who are paid to post online comments with particular content. Thus, this helps the app developers to promote their apps using fake reviews and ratings.

Mostly fraud detecting systems classifies reviews and ratings of the apps into two groups *i.e.*, good and bad. But some reviews and rating are not classified into their relevant groups because of mixed reviews. So, to properly classify apps into their relevant group we use fuzzy logic mechanism which works on all intermediate possibilities between good and bad. The range of possibilities between good and bad includes:

Table 1. Possibilities between good and bad

Very good
Good
Neutral
Bad
Very bad

The fuzzy logic works on the levels of possibilities of input to achieve the definite output. We have used free dataset named App-Review-Dataset of Github repository which has two files namely, *positive10k* and *negative10k*. The positive10k file consists of positive reviews and the negative10k file consist negative reviews of some of the top apps of android app store. ID of the apps whose reviews are taken is stored in the file named as appsid. The total number of apps present in the dataset is 770. The architecture of proposed work is shown in figure 1. The .csv (comma separated values) file of the App-Review-Dataset is used which consist id of the apps, their reviews and ratings. The apps id, reviews and ratings are extracted from the dataset. The ratings of the app are divided into classes. By performing analysis on 80 apps with different threshold value and class value, it is analyzed that for class value 5 and threshold value 0.2, more accurate results are obtained. So, the class value is defined as 5 where number of reviews in each class will not exceed more than 5. The ratings are divided into classes and mean rating of every class is calculated.

 ISSN: 2454-4248
654 – 662

The reviews are converted into ratings using fuzzy logic, the action words are used to classify the reviews to their relevant group. Using fuzzy logic rules, scores are assigned to the reviews. Thus, when score is assigned to review, it gets converted into rating. Furthermore, reviews are divided into classes and mean review of each class is calculated. Theses mean review and mean rating values are used to plot pie chart to detect whether the app is genuine of fake. Variance of mean review and mean rating is calculated. The threshold value for every app is defined as 0.2 to accurately detect whether the app is genuine or fake. The final variance value is compared with the threshold value to check whether the app is fake. If the value of variance is less or equal to the value of threshold, and if the piechart is equally divided then the app is considered as genuine else fake.

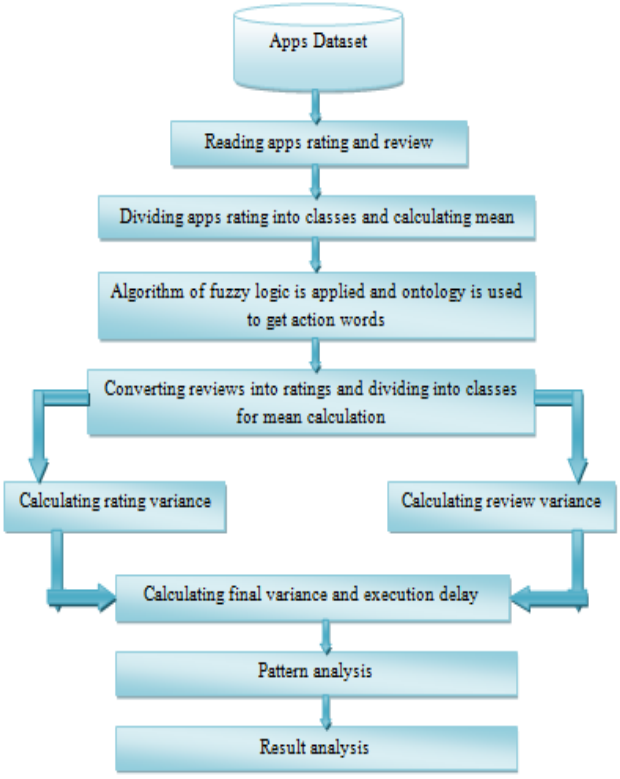


Figure 1.1 Work flow of proposed system

II. ISSUES RELATED TO FRAUD APP DETECTION

The existing fraud app detection system deals with two groups *i.e.*, good or bad to detect whether the app is genuine or fake. Since user's write reviews which can be neutral or mixed should be classified properly. The idea of proposed work is to develop mechanism to resolve the problem to effectively classify the reviews into their relevant groups, in order to get more accurate results. For this, fuzzy logic is used and the accuracy of the results is calculated.

III. OBJECTIVES

The main aim of this project is to design a system which will detect fake apps by considering different evidences indicating their true behavior. We aggregate two types of evidences namely, rating evidences and review evidences. To achieve the aim following Objectives are identified: -

- To find apps are fake or genuine.
- The core objective of this work is to classify the apps using fuzzy logic.
- To improve the differentiation between the reviews.
- To increase classification accuracy of a system.
- Result of this work is used for classification of apps for their verification.

IV. RELATED WORK

Many deceptive behaviors occur in well-known Android app market i.e., Google Play Store. So, to detect malware, previously the work only focused on permission analysis and app executable. Although, Mahmudur Rahman et. al., introduced FairPlay, a system which detected and leveraged trace leave behind by fraudulent to identify the malware and also the apps subjected to search rank fraud [1]. FairPlay correlated review activity and identified their relation with linguistic and behavioral signals that have been gathered from Google Play app data to detect the suspicious apps. Day by day use of mobile has increased. Also to access all types of mobile application, the mobile users prefer to use smartphones. Users generally download mobile applications depending on how many users already have downloaded that application? , what are its ratings and reviews? , what are the comments? etc. Fraud ranking in the app market indicates false or wrong deeds that might have reason to push up apps on the popularity list. Most app developers use fraud means to increase their app's sales by notifying false ratings of the apps, and carrying out ranking fraud. As well as, Varsha A. Patil et. al., presented work on opinion of research on emoticons which is a string of symbols representing different faces in text-based communication [2]. Also an optimization-based aggregation method has been demonstrated, and opinion analysis has been used to find how much a review is positive or negative. The review score has been used to raise the rating score of the user and the emoticons in the reviews or comments. Whereas, Alexis Silva et. al., in [3] introduced a system named BehaviorDroid to track the general properties of apps during run-time. These general properties were currently specified with the help of automata that helped to differentiate between the wanted and undesirable interactions between the resources of phone and app.

 ISSN: 2454-4248
654 – 662

However, in paper [4] Josh Jia Ching Ying et. al., proposed an extremely effective fraud phone call detection approach called parallelized graph-mining, namely PFrauDetector. It automatically labeled deceptive phone numbers by the tag named "fraud", so as to differentiate the fake phone call numbers from the genuine ones. It also used Hyperlink-Induced Topic Search (HITS) algorithm and a novel aggregation approach. In paper [5], Navdeep Singh et. al., presented an optimization-based aggregation approach to incorporate the authentication so as to analyze the capacity of main particular time span from mobile application. Although, Hengshu Zhu et. al., presented a comprehensive approach for ranking fraud and also for detecting ranking fraud in the mobile apps. Firstly, an active period has been mined accurately to locate the ranking fraud [6] [13]. Furthermore, ranking-based evidences, rating-based evidences and review-based evidences were investigated. These three proofs were investigated by representing the apps ranking, rating and review behaviors with the help of statistical hypotheses tests. Also, an optimization-based aggregation method has been proposed that will incorporate entire evidence of fraud detection. The recommended approach was estimated with the real-world app data which was collected from the iOS app store. Spam web pages posed great challenge to the development of search engine. The content spam was commonly used. Along with the development of Internet technologies, the content spam was difficult to detect. For this reason, Jing Wan et. al., [7] has been proposed a detection method for the web page using content spam technique that primarily relies on the statistical features. A spam webpage detection method based on topic and semantics was proposed, with the use of two categories of features, namely, semantics and statistics. To model the information of mobile apps against mobile app services, Hengshu Zhu et. al., presented a sequential approach which is based on hidden markov model (HMM) in [8]. Specifically, first popularity based HMM (PHMM) has been presented to model the sequences of the heterogeneous popularity observations of mobile apps. Also, a method named bipartite-based method has been introduced to precluster the popularity observations. Thus it effectively helped to learn the parameters and initial values of PHMM.

Author Siqi Ma et. al., of paper [9] has been proposed an active and semi-supervised technique to detect the malwares. The approach made use of both known harmless and mischievous apps to predict other malignant apps. The proposed approach was also able to select a good set of apps for experts to label as malicious or harmless to form a set of labeled training data. Furthermore, the approach used both labeled data and unlabeled data, which is a semi-supervised approach. Mayank Taneja et. al., introduced a novel approach for prediction of click fraud in mobile advertising which comprised of feature selection using Recursive Feature Elimination (RFE) and classifies using Hellinger Distance Decision Tree (HDDT) in [10]. RFE was chosen for feature selection as it provided better results as compared to wrapper approach when evaluated using different classifiers. HDDT has been also selected as classifier so as to deal with class imbalance issue present in the data set. The efficiency of proposed framework was investigated on the data set provided by Buzzcity [21] and compared with J48, Rep Tree, logitboost, and random forest.

V. WORK CARRIED OUT

The list of apps is extracted which is present in the form of their ids and each app consists of its positive, negative reviews and ratings. The proposed system considers two parameters to detect fraud app namely, reviews and ratings of apps. Furthermore, concept of class value and threshold value is used. When an app is selected it shows the number of reviews and ratings of the selected app. Also the ratings are divided into classes. The reviews extracted are converted into ratings with the help of following steps:

- Sentence Segmentation
- Convert the sentence into lowercase
- Tokenization
- Remove stop words
- Apply fuzzy logic algorithm

Algorithm for fuzzy logic:

- Split the input into fuzzy sets
- Store If-Then rules
- Simulates reasoning process by making fuzzy inference on input and if-then rules
- Transform the fuzzy set obtained by inference engine into crisp value.

Developed Rules for Fuzzy Logic

- If the counter value of good class is less than bad class and if counter value of bad class is equal to 1, then the score assigned is 2 else 1.
- If counter value of good class is greater than bad class then the score assigned is 3.
- If the counter value of good class is greater than bad class and if counter value of good class is equal to1, then the score assigned is 4 else 5.

After extracting reviews, each sentence is segmented. It is the process of breaking the string text into sentences. This is done by searching for punctuation mark, mostly full stop in English Language. The segmented sentence is converted into lower case and then tokenized. The tokenization is done by searching space after each word. The stop words are removed and action words

 ISSN: 2454-4248
654 – 662

are selected. The action words are compared with the ontologies which are developed. We develop 4 ontologies namely positive ontology (which consist words like fair, awesome, larger, improved, clean, evergreen, best, nice, appropriate, pretty, available, added, excellent, wonderful, better, cooperative, well, large, created, hygience, conducted, fabulous, cheap, facilitated, superb, marvelous, helpful, above, pleasant, qualified, maintained, modified, enough, increased, strict, peaceful, rocks, fine, sweet, high standard, solved) , negative ontology (which consist words like bad, few, shortage, major, limited, crowded, average, worst, bigger, slow, less, adequate, sufficient, missing, no, poor, small, old, terrible, congested, null, rarely, insufficient, expensive, absent, worse, impure, little, unhygience, frequently, boring, difficult, costly), inversion ontology (which consist words like not, would be, should be, needs, must, can be) and more ontology (which consist words like very, extremely, too, more).

Then fuzzy logic algorithm is used which has five groups namely very bad, bad, neutral, good, and very good. Each group is assigned a score from 1 to 5. Thus fuzzy logic algorithm classifies the words into their relevant groups. After converting reviews into ratings, mean review and mean rating of each class is calculated with the help of equation (1)

$$A = \frac{S}{N} \qquad (1)$$

Where,
A= arithmetic mean
N=the number of terms
S=the sum of the number

The mean of each class is use to draw pie chart where pie chart is used for analyzing the phases of each class namely, rising phase, maintaining phase and recession phase.

Furthermore, variance of all class reviews and ratings are calculated with the help of equation (2)

$$\sigma^2=\Sigma(X-\mu)^2/N \qquad (2)$$

Where,
σ = variance
Σ = summation
X = each score (*i.e.*, review or rating)
μ = mean of the score
N = total number of reviews and ratings

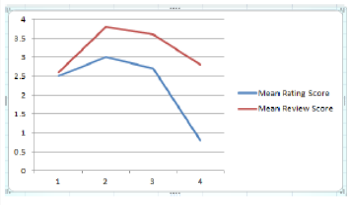
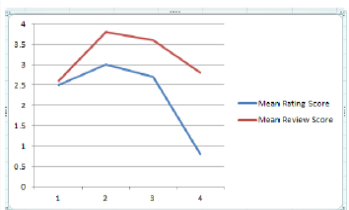
Thus, final variance is calculated by calculating mean of review variance and rating variance. The value of final variance is calculated with threshold value and if the value of final variance is less or equal to the threshold value, the app is considered as genuine else if the value of final variance is greater than the threshold value, the app is considered fake.

VI. EXPERIMENTAL ANALYSIS ON PROPOSED TECHNIQUE

The experimentation of proposed method is carried on App-Review-Dataset of Github repository. The total number of apps present in the dataset is 770.Some apps id, reviews and ratings are added purposely to analyze the robustness of the system under different modifications like threshold value, class value *etc.* The numbers of apps used in experimental analysis are 80. To find which class value and threshold value gives more accurate results with less time requirement, we experimented our approach as shown from figure 2 to 7.

Figure 2, shows the experimental analysis of different class value and threshold value on app id com.deporlovers.copaoro2015 which has 20 number of reviews and ratings. When class value is defined as 5, the number of reviews and ratings get divided into 4 classes where each class consists of 5 reviews and ratings, the graph formed of 4 classes by taking their mean review and rating value is shown in figure 4.2. The threshold value defined is 0.5 which is smaller than the final variance value. So, the result obtained whether the app is genuine or fake by computation is fake which is correct when analyzed by us.

Similarly, when threshold value is defined as 0.4, it is also smaller than final variance value. So the result obtained whether the app is genuine or fake by computation is fake which is correct when analyzed by us. The analysis is carried out by analyzing the variations of flow graph. Thus, in both conditions flow of graph has variations because in case of genuine app, it's phases namely rising phase, maintaining phase and recession phase does not have lot of variations. So we analyze that the computational result is correct.

App id: com.deporlovers.copaoro2015	Sr. no.	No. of review & rating found	Defined class value	Defined threshold value	Total classes formed	Result found = Genuine / Fake	Graph	Analysis (our finding)
	a.	20	5	0.5	4	F		☑
	b.	20	5	0.4	4	F		☑

 ISSN: 2454-4248
654 – 662

Figure 2. Experimental analysis on app id com.deporlovers.copaoro2015

When class value is defined as 4, the number of reviews and ratings gets divided into 5 classes where each class consists of 4 reviews and ratings. The graph is formed of 5 classes by taking their mean review and rating value as shown in figure 3. The threshold value defined is 0.4 which is smaller than the final variance value. The result obtained whether the app is genuine or fake by computation is fake which is correct when analysis is carried out.

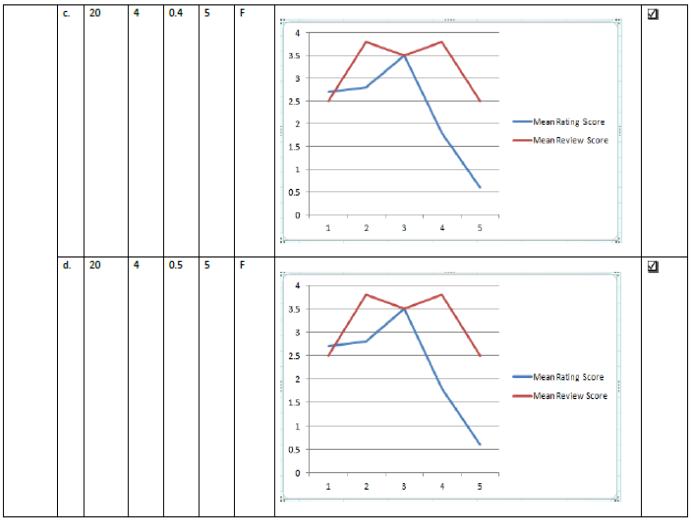


Figure 3 Experimental analysis on app id com.deporlovers.copaoro2015

Similarly, when threshold value is 0.5, it is smaller than the final variance value. The result obtained whether the app is genuine or fake by computation is fake which is correct when analysis is carried out.

In figure 4, for class value 4 and threshold value 0.3, the number of classes formed are 5 each containing 4 reviews and ratings. The result obtained whether the app is genuine or fake by computation is fake which is correct when analysis is carried out.

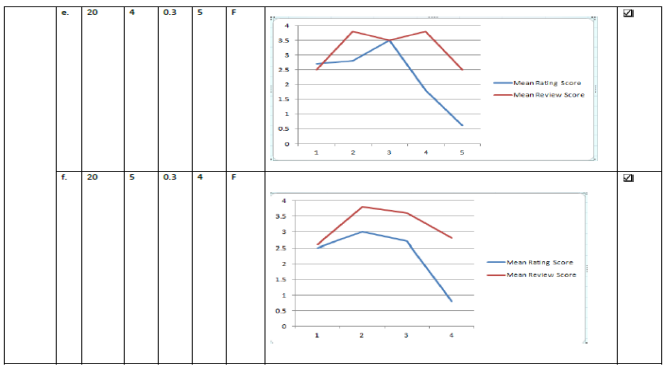


Figure 4. Experimental analysis on app id com.deporlovers.copaoro2015

For class value 5 and threshold value 0.3, the numbers of classes formed are 4. The result obtained whether the app is genuine or fake by computation is fake which is correct when analysis is carried out.

When the threshold value is 0.2 and class value is 4, the numbers of classes formed are 5. In figure 5, the result obtained whether the app is genuine or fake by computation is fake which is correct when analysis is carried out. For class value 5 and threshold value 0.2 in figure 4.5, the result obtained is fake by computation which is correct when analyzed by us.

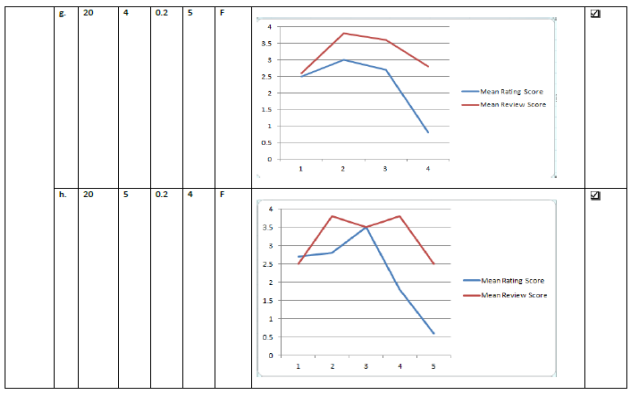


Figure 5 Experimental analysis on app id com.deporlovers.copaoro2015

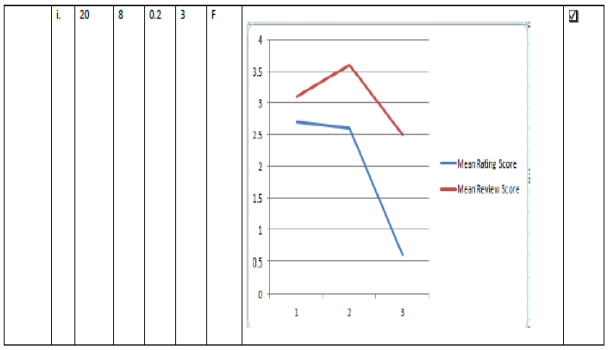


Figure 6 Experimental analysis on app id com.deporlovers.copaoro2015

For class value 8 and threshold value 0.2, the numbers of classes formed are 3 as shown in figure 6. The result obtained by computation is fake because the value of final variance is greater than threshold value which is correct when analysis is carried out.

Similarly, we experimented our proposed approach on 80 apps which gives results as follows, the figure 7 shows number of accurate and inaccurate classification of 80 apps. As from 80 apps, 75 apps are accurately classified with class value 5 and threshold value 0.2 we selected them as class value and threshold value for our proposed approach.

ISSN: 2454-4248
654 – 662

A	B	C	D
		Experimental Analysis on 80 apps	
Class Value	Threshold value	Accurate classification	Inaccurate classification
5	0.5	19	61
5	0.4	37	43
4	0.4	42	38
4	0.5	28	52
4	0.3	62	18
5	0.3	57	23
4	0.2	73	7
5	0.2	75	5
8	0.2	21	59

Figure 7 Result of experimental analysis on 80 apps

VII. OUTPUT OF PROPOSED TECHNIQUE

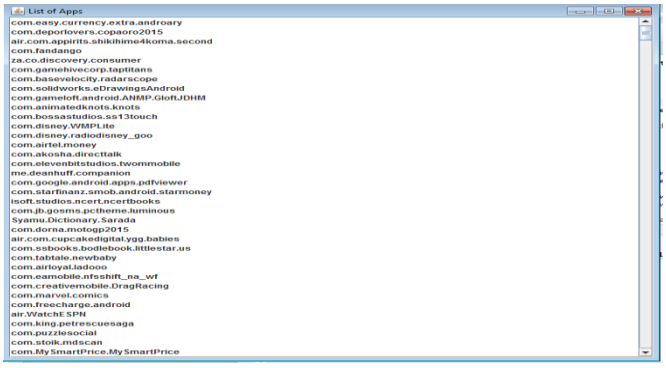


Figure 8: List of Apps

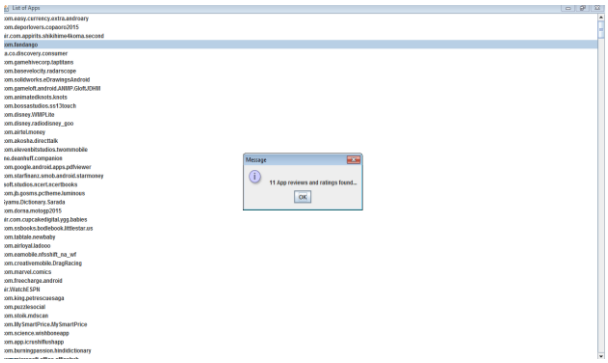


Figure 9: Number of reviews found for app id com.fandango.

Figure 8 shows list of apps to be used. When an app is selected, it shows the number of reviews and ratings of the selected app. In figure 9, app with id com.fandango is selected and it shows 11 reviews and ratings of that app is found.

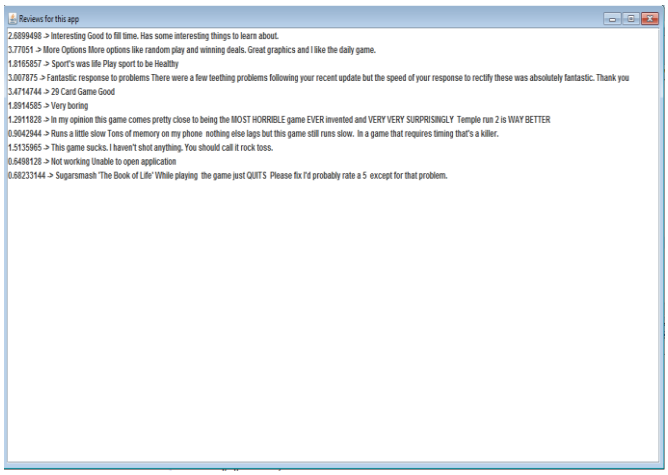


Figure 10: - Reviews and ratings of the app com.fandango

As shown in figure 10, when clicked on ok button, the reviews and ratings of selected app *i.e.*, com.fandango is made known.

Figure 11 shows the analysis of app whether it is fake or genuine. As 11 reviews and ratings are found, each class consists of 5 reviews and ratings because class value is defined as 5. So, the number of classes formed is 3. Thus, ratings of the app are divided into classes and mean rating of each class is calculated. The reviews of the app are converted into ratings and mean review of each class is calculated. Furthermore, variance of review and rating is calculated followed by final variance. The app is stated as fake because the value of final variance is 0.6291424137662998 and the threshold value is defined as 0.2 which is less than final variance value.

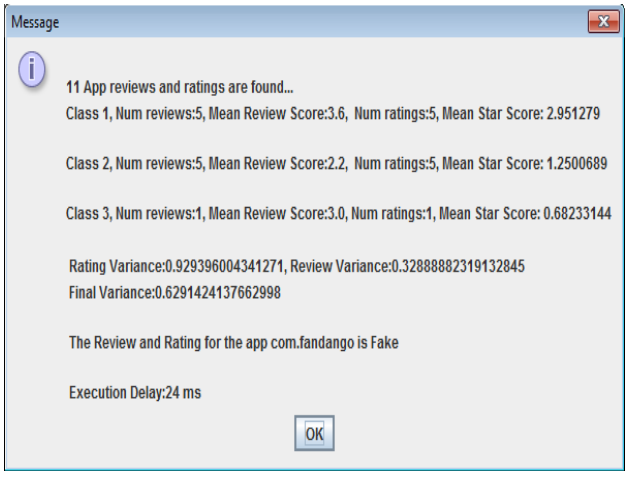


Figure 11: - Analyzing app com.fandango is genuine or fake

Figure 12 illustrates the mean review score of app id com.fandango. The red color in piechart depicts mean review value of class 1.

ISSN: 2454-4248
654 – 662

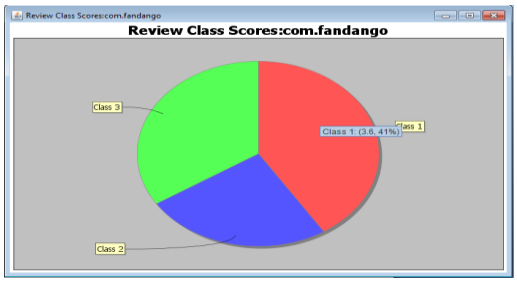


Figure 12: - Mean review score of class 1

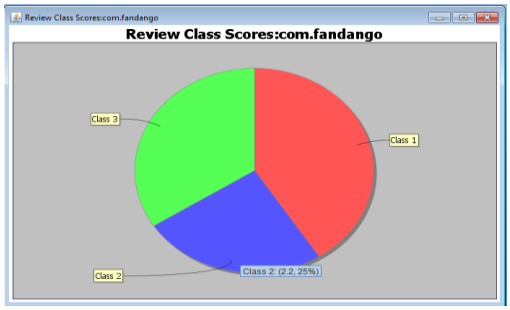


Figure 13: - Mean review score of class 2

The dark blue color in piechart of figure 13 depicts mean review value of class 2. The green color of piechart of figure 14 depicts mean review value of class 3.

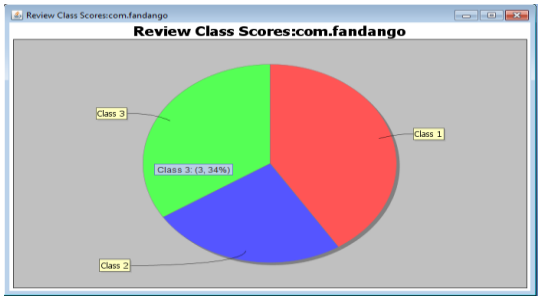


Figure 14: - Mean review score of class 3

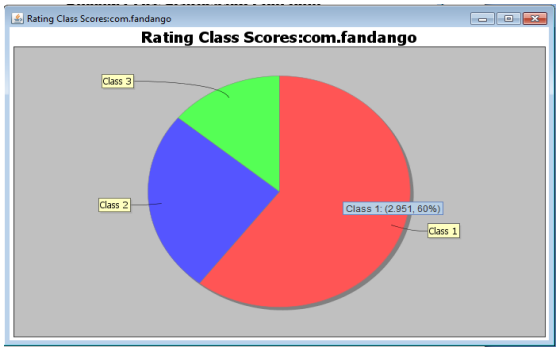


Figure 15: - Mean rating score of class 1

The red color of piechart in figure 15 depicts mean rating value of class 1.

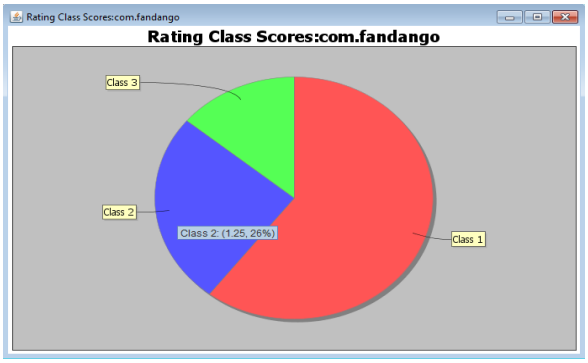


Figure 16: - Mean rating score of class 2

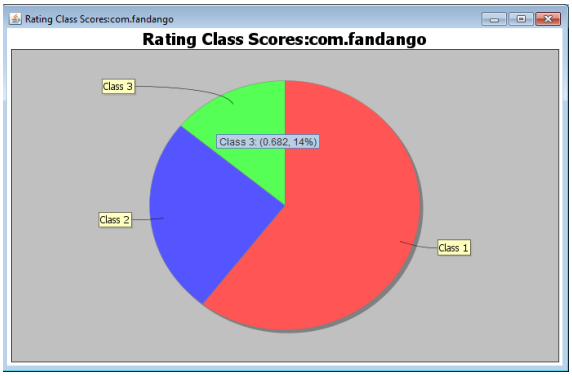


Figure 17: - Mean rating score of class

The dark blue color of piechart in figure 16 depicts mean rating score of class 2. The green color of piechart in figure 17 depicts mean rating value of class 3. From above piecharts we analyzed that its behavior is dissimilar *i.e.*, the rising phase, maintaining phase and recession phase is very divergent, so we analyzed that the app is fake.

VIII. ANALYSIS

During experimentation we studied the variation of and time taken and accuracy for retrieval by apps.

A. Time Analysis

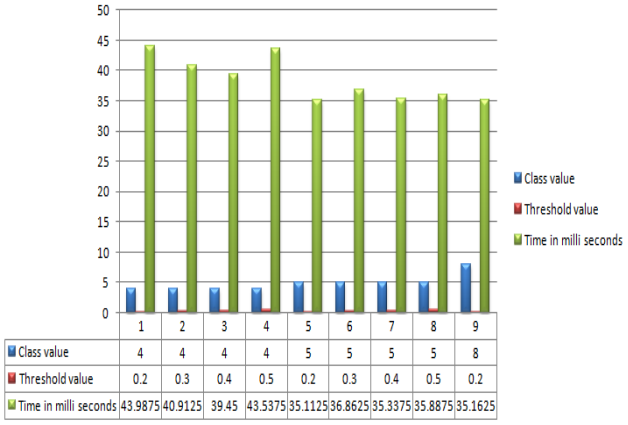
The system is evaluated to check the mean time required for execution of results of apps with different class value and threshold value.

 ISSN: 2454-4248
654 – 662

Class value	Threshold value	Time in milli seconds
4	0.2	43.9875
4	0.3	40.9125
4	0.4	39.45
4	0.5	43.5375
5	0.2	35.1125
5	0.3	36.8625
5	0.4	35.3375
5	0.5	35.8875
8	0.2	35.1625

Figure 18 Time analysis of apps

Figure 18 shows the average time required by all apps with different threshold value and class value. The minimum time required by apps is 35.11.25 with class value 5 and threshold value 0.2. Figure 19 is the graphical analysis of time requirement by all apps with different class value and threshold value.



	1	2	3	4	5	6	7	8	9
Class value	4	4	4	4	5	5	5	5	8
Threshold value	0.2	0.3	0.4	0.5	0.2	0.3	0.4	0.5	0.2
Time in milli seconds	43.9875	40.9125	39.45	43.5375	35.1125	36.8625	35.3375	35.8875	35.1625

Figure 19 Graphical analysis of time taken by apps

B. ACCURACY

Accuracy is defined as the ratio of number of genuine apps that are extracted correctly, to the total number of apps used for experimentation. So, we calculate accuracy with the help of equation (3)

$$Accuracy = \frac{Number\ of\ genuine\ apps\ correctly\ classified}{Total\ no.of\ apps\ used\ for\ experimentation} * 100 \quad (3)$$

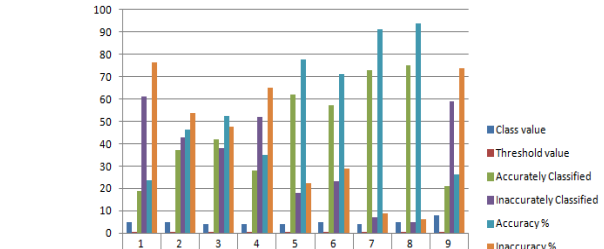
Table 2 shows computation of apps that are accurately classified and inaccurately classified. It is analyzed that with class value 5 and threshold value 0.2, better accuracy is obtained as compared to others.

Table 2 Computation of accuracy

Class value	Threshold value	Accurately Classified	Inaccurately Classified	Accuracy %	Inaccuracy %
5	0.5	19	61	23.75	76.25
5	0.4	37	43	46.25	53.75
4	0.4	42	38	52.5	47.5
4	0.5	28	52	35	65
4	0.3	62	18	77.5	22.5
5	0.3	57	23	71.25	28.75
4	0.2	73	7	91.25	8.75
5	**0.2**	**75**	**5**	**93.75**	**6.25**
8	0.2	21	59	26.25	73.75

Figure 20 shows graph analysis of accuracy with different class value and threshold value. It also shows that for which class value and threshold value the apps are accurately and inaccurately classified.

Thus, it is analyzed that for class value 5 and threshold value 0.2 accuracy of 93.75% is obtained when experimental analysis is carried on 80 apps.



	1	2	3	4	5	6	7	8	9
Class value	5	5	4	4	4	5	4	5	8
Threshold value	0.5	0.4	0.4	0.5	0.3	0.3	0.2	0.2	0.2
Accurately Classified	19	37	42	28	62	57	73	75	21
Inaccurately Classified	61	43	38	52	18	23	7	5	59
Accuracy %	23.75	46.25	52.5	35	77.5	71.25	91.25	93.75	26.25
Inaccuracy %	76.25	53.75	47.5	65	22.5	28.75	8.75	6.25	73.75

Figure 20 Accuracy

IX. CONCLUSION AND FUTURE SCOPE

The main objective of the proposed work was to study fraud detection from apps and to use fuzzy logic approach to differentiate the actual fraud apps. The proposed system performed classification of apps and detect their groups whether they belong to very bad, bad, neutral, good and very good. The experimental analysis is carried on 80 apps with the proposed method for detection of fake apps. Different class value and threshold value gives different results of accuracy of time required for execution. Through analysis, we found that the proposed method when used with class value of 5 and threshold value 0.2 gives accuracy of 93.75% when compared with other class value and threshold value.

 ISSN: 2454-4248
654 – 662

The work can be expanded in future by considering other parameters such as ranking of the app, etc. In future, to increase accuracy of system, new approaches can be used different dataset.

REFERENCES

[1] 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, pp. 1329-1342.

[2] Varsha A. Patil and Nitin N. Patil, "*Mobile Apps Opinion Analysis using Emoticon*", IEEE International Conference on Global Trends in Signal Processing, Information Computing and Communication, 2016, pp. 203-207.

[3] Alexis Silva and Jocelyn Simmonds, "*Behaviordroid: Monitoring Android Applications*", IEEE/ACM International Conference on Mobile Software Engineering and Systems, 2016, pp. 19-20.

[4] Josh Jia-Ching Ying, Ji Zhang, Che-Wei Huang, Kuan-Ta Chen and Vincent S. Tseng, "*PFrauDetector: A Parallelized Graph Mining Approach for Efficient Fraudulent Phone Call Detection*", IEEE International Conference on Parallel and Distributed Systems, 2016, pp. 1059-1066.

[5] Navdeep Singh, Prashant Kr. Pandey and Mr. Srinivasan, "*Improved Discovery of Rating Fake for Cellular Apps*", IEEE International Conference on Science Technology Engineering and Management (ICONSTEM), 2016, pp. 135-140.

[6] Hengshu Zhu, Hui Xiong, Yong Ge, and Enhong Chen, "*Discovery of Ranking Fraud for Mobile Apps*", IEEE Transactions on Knowledge and Data Engineering, Vol. 27, No. 1, January 2015, pp. 74-87.

[7] Jing Wan, Mufan Liu, Junkai Yi and Xuechao Zhang, "*Detecting Spam Webpages through Topic and Semantics Analysis*", IEEE Global Summit on Computer and Information Technology (GSCIT), 2015, pp. 83-92.

[8] Hengshu Zhu, Chuanren Liu, Yong Ge, Hui Xiong and Enhong Chen, "*Popularity Modeling for Mobile Apps: A Sequential Approach*", IEEE Transactions on Cybernetics, Vol. 45, No. 7, July 2015, pp. 1303-1314.

[9] Siqi Ma, Shaowei Wang, David Lo, Robert Huijie Deng, and Cong Sun, "*Active Semi-supervised Approach for Checking App Behavior against its Description*", IEEE Annual International Computers, Software & Applications Conference, 2015, pp. 179-184.

[10] Mayank Taneja, Kavyanshi Garg, Archana Purwar and Samarth Sharma, "*Prediction of Click Frauds in Mobile Advertising*", IEEE International Conference on Contemporary Computing (IC3), 2015, pp. 162-166.

[11] Kaiyu Wang, Yumei Wang, Hongqiao Li, Yilin Xiong and Xinyu Zhang, "*A New Approach for Detecting Spam Microblogs Based on Text and User's Social Network Features*", IEEE International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronics Systems (VITAE), 2014, pp. 1-5.

[12] Hengshu Zhu, Enhong Chen, Hui Xiong, Huanhuan Cao, and Jilei Tian, "*Mobile App Classification with Enriched Contextual Information*", IEEE Transaction on Mobile Computing, Vol. 13, No. 7, Jul. 2014, pp. 1550–1563.

[13] Hengshu Zhu, Hui Xiong, Yong Ge and Enhong Chen, "*Ranking Fraud Detection for Mobile Apps: A Holistic View*", ACM International Conference on Information Knowledge Management (CIKM), 2013, pp.619–628.

[14] Hengshu Zhu, Enhong Chen, Kuifei Yu, Huanhuan Cao, Hui Xiong and Jilei Tian, "*Mining Personal Context-Aware Preferences for Mobile Users*", IEEE International Conference on Data Mining (ICDM), 2012, pp.1212–1217.

[15] Sihong Xie, Guan Wang, Shuyang Lin, and Philip S. Yu, "*Review Spam Detection via Temporal Pattern Discovery*", ACM International Conference on Knowledge Discovery Data Mining (KDD), 2012, pp. 823–831.

[16] Hengshu Zhu, Enhong Chen, Kuifei Yu, Huanhuan Cao, Hui Xiong and Jilei Tian, "*Mining Personal Context-Aware Preferences for Mobile Users*", IEEE International Conference on Data Mining (ICDM), 2012, pp.1212–1217.

[17] Bo Yan and Guanling Chen, "*AppJoy: Personalized Mobile Application Discovery*", ACM International Conference on Mobile Systems Applications and Services (MobiSys), 2011, pp. 113–126.

[18] The mentioned statistics of google play store. Online.. Available: https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/

[19] The mentioned statistics of apple's app store. Online.. Available: https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/

[20] .20. The mentioned dataset App-Review-Dataset. Online.. Available: https://github.com/amitt001/Android-App-Reviews-Dataset

[21] 21. Buzzcity dataset. Online.. Available: https://larc.smu.edu.sg/buzzcity-mobile-advertisement-datas