International Journal of Modern Trends in Engineering and Research
www.ijmter.com
e-ISSN No.:2349-9745, Date: 2-4 July, 2015

Using Enriched Contextual Information of App Context for Risk Score Based App Classification

Miss. Prajakta Lokhande[1], Mr. Shivaji Lahane[2]

[1]*Department of Computer Engineering, G.E.S's. R.H. Sapat College of Engineering,* Nasik
[2]*Department of Computer Engineering, G.E.S's. R.H. Sapat College of Engineering, Nasik*

Abstract- Large numbers of mobile apps with similar functionality are coming into the market due to the increasing use of mobile devices. Classifying these apps is useful to understand the user preferences which can motivate the intelligent personalized services and also while selecting the application. But turns out be a nontrivial task as limited information about the apps is directly available. In this paper we have presented a method to classify the mobile apps using the information collected from the various sources like information from apps name, search engine, contextual usage history collected from the user's usage record and also the permissions of the app. This will provide us a secure and effective classification of the apps as most of these apps come from an unknown vendor and so there is higher possibility of them being malicious.
Keywords- Mobile apps, Classification, Security, Risk score, Enriched information.

I. INTRODUCTION

Today the mobile devices have turned into a ubiquitous device, due to the evolution in technology providing various facilities like camera, email, video calling access to social network, etc. As a result large numbers of mobile apps, most of them having similar functionality are coming into the market. Proper classification of these apps can help user in order to search the required app easily and also give a proper mobile app usage analysis. This can motivate different intellectual services like app recommendation, target advertising, user segmentation etc [1], is considered as a quite difficult task, because for having a proper or effective classification we need to have detailed information about the app. This is challenging task as the contextual information available about the app is very limited this is because the words used for app name are very short and sparse.

Also it is found that the current security mechanism provided by android cannot detect such malicious apps, as the security mechanism provided by android is in standalone fashion [2] i.e. the user is supposed to make the decision about accepting app, by allowing the app to access the resources requested after reading the permissions requested by that app, which is easily ignored by the users. As a solution to these problems a system is being proposed a system that will provide an effective classification of the mobile apps along with their risk score by using the enriched information about the apps. The enriched information here consist of the information of the apps obtained from various sources like name of the app, web based information, the real world contextual features about the app and list permissions requested by the app at the time of installation, providing us a more accurate and secure classification of the mobile apps.

II. LITRETURE SURVEY

Our problem to automatically classify the mobile apps can also be considered as a problem to classify the short and sparse text. Here we have considered the existing work done first with regards to the classification of the short and sparse texts and then the earlier work done for improving the security of the android system.

2.1 Classification of Short and Sparse Text

X. H. Phan et al [3] presented a general framework to process the short and sparse text documents on the web. They have focused mainly on the data sparseness and synonyms/hyponyms by exploiting the hidden topics discovered from large scale external document collection; this helped to improve the representation of the short and sparse text for the classification. Sahami and T.D. Heilman [4], presented a similarity kernel function based approach for finding the similarity between the short text and proved that there approach can effectively measure the similarity between short text snippets. A.Z. Broder et al [5] proposed a methodology to classify these short queries using blind feedback technique. Authors proved that their methodology yields higher classification for the queries with help of the empirical evaluation performed. H. Ma, et al [6], proposed an approach which leverages search snippets to build vector space for both app usage and categories and classify the app usage records using the cosine space distance. H. Zhu, H et al [7] proposed an approach to classify the mobile app by first exploited the web based features of the app from the web search engine, but as the information obtained is limited they then also considered the explicit contextual information obtained from the mobile users context-rich device logs.

2.2 Android Security

Applications before installation ask for the permissions to access some or the other kind of information from user's device. Unknown to the user these apps may get access to some sensitive information present on the users device, which might be harmful to the user in case the malicious apps uses this information for their beneficial purpose. In order to make the user aware about what kind of data is being accessed by the apps they have installed, W.Enack et al [7], proposed one tracking system for having a real time privacy monitoring on the smart phones that informs user when the application is trying to send sensitive data from the phone. A.P. Felt et al [9] studied the permissions the apps request at the time of installation and came to conclusion that most the apps asked for large number of permissions, which they don't even require for their processing. This makes the apps more threatening as the attacker through these apps may try to get access to the sensitive information of the user. Authors here have used the static analysis to determine the over privileged apps, on a set of 940 applications and find that about one-third are over privileged. E. Chin et al [10] conducted one user study to understand user's perceptions of smart phone security and installation habits. Where they found that the users don't focus on the permissions at the time of app browsing and installation, they mostly relay on the user rating and reviews while selecting the app. B.P. Sarma et al, [11] investigated the feasibility of apps permissions with respect to their category without considering the usefulness of the app.

III. PROPOSED SYSTEM

Fig. 1 shows the system architecture of proposed system, it consists of a mobile application for user that provides user interface to the user and the whole app classification process is carried out at the server end. First for classifying the mobile apps, the information is collected and exploited from various sources like information from the labels (app name), web based features information collected from the search engine; contextual features information collected from the mobile app used by the volunteers specially designed for monitoring and storing this information at the server and the list of permissions extracted from the manifest file of the apps. Based on these features the apps will be classified using the machine learning technique. After the Classification results are obtained they are stored in the database. Administrator is present to manage the system. The process of classification using theses different information is carried out in different phases as shown in the Fig. 2.

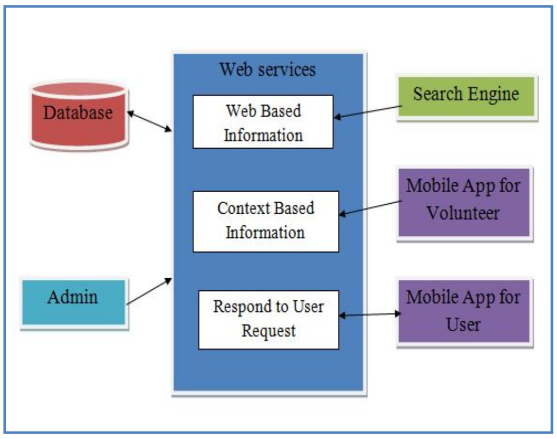


Fig.1 System Architecture

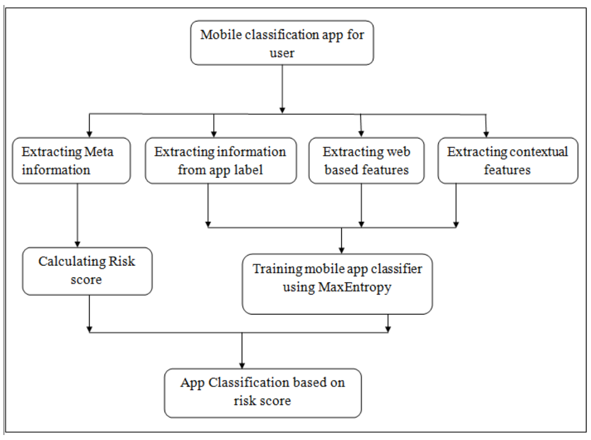

Mobile classification app for user
Extracting Meta information
Extracting information from app label
Extracting web based features
Extracting contextual features
Calculating Risk score
Training mobile app classifier using MaxEntropy
App Classification based on risk score

Fig.2 Block Diagram of Different Phases of the system

3.1 Phase 1: Feature extraction

In this phase features of the app will be extracted from different sources. Our system considers both the explicit and implicit features of the app as explained below:

3.1.1Web based feature extraction:

Here the features extraction takes place with the information obtained from the web search engine. Both the explicit and implicit features are obtained as explained below.

3.1.1.1 Explicit web based feature extraction:

Here the explicit features of the app will be extracted from web search engine. In simple words after giving the name of the app to the system, it will extract the top results from the search engine to place the app in the appropriate category. To achieve this goal vector space model will be used. Which consist of the three steps, according to which first, one category profile d_c will be build by integrating all the M snippets retrieved for some app. Here the stop words will be removed and the verbs and adjectives will be normalized Then in the second step normalized word vector will be build for each app category $\overrightarrow{w_c}$ = dim [n] using the following formula [1]:

$$\text{dim}[i] = \frac{freq_{i,c}}{\sum_i freq_{i,c}} \quad (1 \leq i \leq n) \qquad (1)$$

Where, $freq_{i,\,c}$ is the frequency of ith word in the category profile. Similarly, in the last step for each snippet s retrieved for the app a we will build word vector $\overrightarrow{w_{a,s}}$ and calculate the cosine distance between $\overrightarrow{w_c}$ and $\overrightarrow{w_{a,s}}$. From the result obtained, we will consider the max similarity result and assume that category to be the appropriate category for app a.

$$c^* = \text{arg max Similarity} \left(\overrightarrow{w_{a,s}}, \overrightarrow{w_c} \right) \qquad (2)$$

Then to confirm our result we calculate the general label confidence score by [1]:

$$GConf(c, a) = \frac{M_{c,a}}{M} \qquad (3)$$

Where, $M_{c,a}$ is the number of returned related search snippets of app a, whose category labels are c after mapping.

3.1.1.2 Implicit web based features extraction:

Here the latent semantic meaning of the snippets retrieved will be considered giving us a more refined result. This is because in the explicit feature selection the latent semantic meaning of the words is not considered i.e. for example the words like ``play", ``game" and ``funny" are considered as totally different while calculating the distance between the word vectors. But after considering their latent semantic they can be placed in the same semantic topic of ``entertainment". To understand the latent semantic meaning we will use the latent dirichlet allocation (LDA) model. Using this model we will build a category profile d_c. Then for given app a top M result will be retrieved and the words which do not match with category profile will be removed. Then KL divergent will be used to map each snippet with the category. The formula for KL-Divergent is given by [1]:

$$D_{KL}(P(z|s)||P(z|c)) = \sum_k P(z_k|s) ln \frac{P(z_k|s)}{P(z_k|c)} \qquad (4)$$

Where, z is the latent topic of the category. The category with the smallest KL-divergence is selected. To confirm our result we then calculate the topic confidence score for the given category as follows [1]:

$$TConf(a,c) = \frac{T_{a,c}}{M} \qquad (5)$$

Where, $T_{a,c}$ is the number of returned snippets of app a with respect to category label c. The topic confidence score gives the confidence that app a is labelled as category c with respect to the latent semantic topic.

3.1.2 Contextual Feature Extraction:

In this we consider the context feature of the app from real world context log. Here also we consider the explicit and implicit features.

3.1.2.1 Explicit Contextual Feature :

Here the real world context information is collected from different user's logs in terms of feature-value pair. In simple words for an app belonging to some category we consider when the app is being used and how many time it is being used. A context profile R_a is build by using these feature-value pair, for each app. Similarly we also build a context profile for each category R_c, by combining the context profiles of the pre-selected apps labeled with c. After that a vector is build for the app and category and we calculate the cosine distance between the context vector of the app and context vector of all the categories. At last the cosine distance calculated is arranged in descending order and the one with max similarity is assumed as the category. The result is then confirmed by calculating category rank distance given by [1]:

$$CRDistance\ (a,c) = Rk(c) - 1 \qquad (6)$$

Where, Rk(c) is the rank of the category c, which is obtained by comparing the vector distance to app a. Here smaller the distance more accurate is our selected category.

3.1.2.2 Implicit Contextual feature:

In implicit feedback the semantic meaning of the contextual feature-value pairs will be obtained from the explicit contextual features available. Like for example feature-value pairs like "Time-span: 2 hours", "age: 15-20" can be grouped together under the topic "trendy app". To achieve this goal here we category profile R_c is build using Latent Dirichlet Allocation(LDA) on context model. Accordingly, for a given app a, category profile Ra will be build using the historic logs database. Then KL-Divergence will be calculated between each category and the app using the given formula [1]

$$D_{KL}(P\ (z|\ Ra)\ ||\ P\ (z\ |\ Rc)) = \sum_k P\ (z_k\ |Ra)\ ln \frac{P(z_k|Ra)}{P(Z_k|Rc} \qquad (7)$$

From this the smallest KL-Distance is considered as the category for the app and finally, topical Rank distance will be calculated to confirm the result. The topical rank distance is calculated by:

$$TRDistance\ (a,c) = Rk(c) - 1 \qquad (8)$$

3.2 Phase 2: Training the Classifier

After the extraction of the features, classifier will be trained to classify the app into its appropriate category by combing these different features. From the different classifier present we have considered the MaxEntropy classifier as it gives best results when the information available is sparse and it is easy to incorporate different features using this classifier. For that we have used MaxEntropy classifier which is calculated by the given formula:

$$P\,(c|\,d) = \frac{1}{Z(d)}\ \exp\left(\sum_i \lambda i f i(d,c)\right) \qquad (9)$$

Where, $fi(d,c)$ is a feature, Z (d) is a normalizing factor and λi is a parameter to estimate.

3.3 Phase 3: Extracting Meta Information of the App

To make the classification of the system more effective and to improve the security issues of the mobile apps, we use the Meta information of mobile apps i.e. the permissions used by the mobile apps are considered. For this we extract the manifest file attached with each app, from this we get the list permissions the app requests at the time of installation. Then using the rarity of the critical permission we will calculate the risk of the app in a simple user-friendly manner.

3.4 Phase 4: Risk Score Based App Classification:

The earlier stage calculates the app category and risk score of an application. In this stage we provide app classification based on risk score i.e. Apps in each category are ranked in decreasing order as per the risk score. We can also have the comparison of the working of the two classifiers for classification.

IV. EXPERIMENTAL SETUP

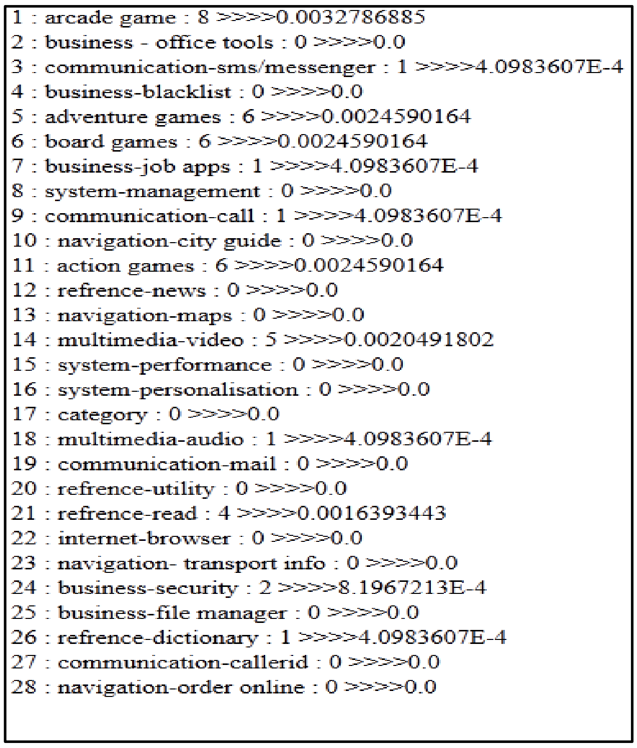
Our system consists of web services and mobile application created using java, apache server and android sdk in eclipse respectively. The mobile apps communicate with the web services using the http protocol and the data transfer between them is carried out using JSON. The apk created for the data collection will provide the information to the context based web service. The user app and the user response web service communicate with each other, in which the user will get the analytical result of the app requested by the user. Administrator is present to maintain the server system. All the results obtained are stored in the database.

Data Sets:

- Context Log: This dataset contains user specific app access information in the form <uid, app name, access details>
- App Category: We have created 2 level category set. The level-1 contains 9 category lists. Each category is the subdivided in to level-2 categories. We have crated 26 such categories.
- Permission Dataset: This data set contains level-2 category wise permission details. We create this dataset programmatically. This can be recorded as: <catid, permission list >
- List of Category keyword : this list contains category level-2 specific keyword list <catid, keyword list>

V. RESULTS

Using the dataset we have obtained the results for various apps belonging to the different category. The result obtained using web based feature is shown in the fig. 3 along with its graph as shown in fig.4. We have shown here the result for the angry bird game application. It shows with how much accuracy the app belongs to the arcade category which is a sub category of game.



```
1 : arcade game : 8 >>>>0.0032786885
2 : business - office tools : 0 >>>>0.0
3 : communication-sms/messenger : 1 >>>>4.0983607E-4
4 : business-blacklist : 0 >>>>0.0
5 : adventure games : 6 >>>>0.0024590164
6 : board games : 6 >>>>0.0024590164
7 : business-job apps : 1 >>>>4.0983607E-4
8 : system-management : 0 >>>>0.0
9 : communication-call : 1 >>>>4.0983607E-4
10 : navigation-city guide : 0 >>>>0.0
11 : action games : 6 >>>>0.0024590164
12 : refrence-news : 0 >>>>0.0
13 : navigation-maps : 0 >>>>0.0
14 : multimedia-video : 5 >>>>0.0020491802
15 : system-performance : 0 >>>>0.0
16 : system-personalisation : 0 >>>>0.0
17 : category : 0 >>>>0.0
18 : multimedia-audio : 1 >>>>4.0983607E-4
19 : communication-mail : 0 >>>>0.0
20 : refrence-utility : 0 >>>>0.0
21 : refrence-read : 4 >>>>0.0016393443
22 : internet-browser : 0 >>>>0.0
23 : navigation- transport info : 0 >>>>0.0
24 : business-security : 2 >>>>8.1967213E-4
25 : business-file manager : 0 >>>>0.0
26 : refrence-dictionary : 1 >>>>4.0983607E-4
27 : communication-callerid : 0 >>>>0.0
28 : navigation-order online : 0 >>>>0.0
```

***Fig.3 Classification using Web based Feature* For Angry Bird app**

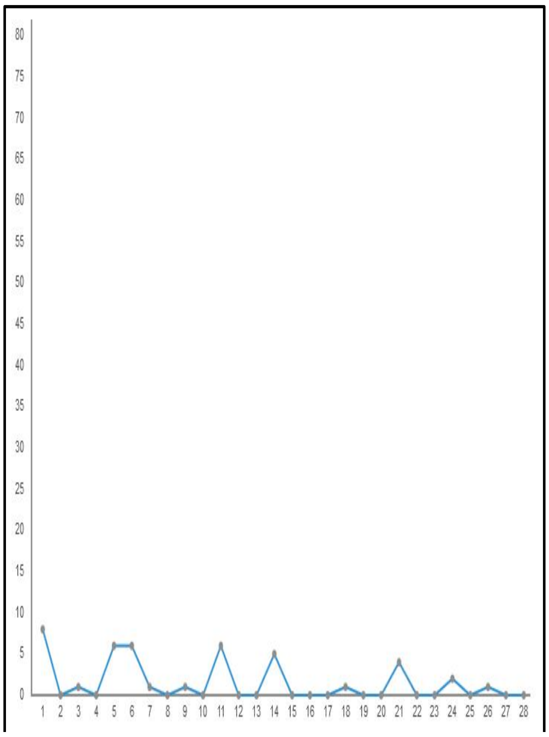


***Fig.4* Graph for Classification using Web based Feature for Angry Bird app**

CONCLUSION

We have classified the mobile apps by using the information obtained from different sources. We have created multiple sub categories which are currently not available in the play store, giving us the more accurate classification of the apps By using the permissions requested by the apps we calculate the risk factor of the apps, informing the user the about how much risky the app will be; thus improving the security concerns of the malicious apps in easy to understand manner. This will not only help the user to select the proper app according to his requirements but also can be used for various other purposes like target advertising, user segmentation for market analysis etc.

REFERENCES

[1] H. Zhu,E. Chen,H. Xiong and H. Cao,``Mobile App Classification with Enriched Contextual Information," IEEE Transactions on mobile computing,Volume:13 , Issue: 07 ,7 July 2014

[2] Christopher S. Gates, et.al,``Generating Summary Risk Scores For Mobile Applications",IEEE Transactions on dependable and secure computing, (Volume :11, Issue: 03), May-June 2014.

[3] X.-H. Phan et al.,``A hidden topic-based framework toward building applications with short web documents," IEEE Trans. Knowl.Data Eng., vol. 23, no. 7, pp. 961–976, Jul. 2010

[4] M. Sahami and T. D. Heilman,``A web-based kernel function for measuring the similarity of short text snippets," in Proc. WWW, Edinburgh, U.K., pp. 377–386,2006.

[5] Z. Broder et al,``Robust classification of rare queries using web knowledge,"in Proc. SIGIR, Amsterdam, Netherlands,pp. 231–238,2007

[6] H. Ma, H. Cao, Q. Yang, E. Chen, and J. Tian,``A habit mining approach for discovering similar mobile users,"in Proc. WWW, Lyon, France, pp. 231–240,2012

[7] H. Zhu, H. Cao, E. Chen, H. Xiong, and J. Tian,``Exploiting enriched contextual information for mobile app classification,"in Proc. CIKM,Maui, HI, USA, pp. 16171621,2012.

[8] W. Enck, P. Gilbert, B. Chun, L.P. Cox, J. Jung, P. McDaniel, and A.N Sheth,``Taint- Droid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones," Proc. Ninth USENIX Conf. Operating Systems Design and Implementation, article 1-6,2010
[9] A.P. Felt, E. Chin, S. Hanna, D. Song, and D. Wagner,``AndroidPermissionsDemystified,"Proc. 18th ACM Conf. Computer and Comm. Security, pp. 627-638, 2011.
[10] E. Chin, A.P. Felt, V. Sekar, and D. Wagner,``Measuring User Confidence in Smartphone Security and Privacy," Proc. Eighth Symp. Usable Privacy and Security, (SOUPS '12), article 1, 2012.
[11] B.P. Sarma et al.,``Android Permissions:A Perspective Combining Risks and Benefits," Proc. 17th ACM Symp. Access Control Models and Technologies (SACMAT 12), 2012.
[12] William M. Darling,``A theoritical and practical implementaion Tutorial on Topic Modeling and Gibbs Sampling,"December 1, 2011
[13] NaïveBayesClassification, http://www.saedsayad.com/naive_bayesian.html
[14] K. Nigam et al.,``Using Maximum Entropy for Text Classification,"IJCAI Workshop Machine Learning for Information Filtering,1999,pp.61-67