IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 9, September 2016

Detecting and Classifying Malicious Apps Using Rule Mining Algorithms

Shyam Chandran P[1], Mubeena V[2]

Assistant Professor, Department of Computer Science, Sree Narayana Guru College, Coimbatore, Tamil Nadu, India[1]

M.Phil Scholar, Department of Computer Science, Sree Narayana Guru College, Coimbatore, Tamil Nadu, India[2]

Abstract: In the recent trend, there are plenty of apps dominating social networks. There are several mobile apps and social site based apps are malicious and duplicate in features. Facing the large amount of apps, app retrieval and app recommendation become imperative task, while users can easily use them to acquire their desired apps without malicious and duplicate. To classify different apps based on its features and security levels is the major task of the proposed work. The recent methods are conducted mostly relying on user's log or app's details, which can only detect whether two apps are downloaded or used by the user. Moreover, apps contain many general relationships other than similarity, such as one app needs many permissions, the proposed work classifies the app based on the permission asked by the app. These relationships cannot be performed without the whole details of app descriptions. Reviews contain user's viewpoint and judgment to apps, thus they can be used to calculate relationship between apps. To use reviews, this paper proposes a similarity and rule matching process by combining review similarity and app rule verification.

Keywords: Data Mining, Mobile APP, Relations among complexity measures, similarity measures, text processing, web mining, malicious detection.

I. INTRODUCTION

In the past few years, smart phone becomes essential part of life. The smart phones are used for different purposes such as education, entertainment, trading, and travel, etc. Mobile apps are software applications designed to run on smart phones, tablet computers and other mobile devices.The popularity of smart phones causes many apps released to help users make the best use of their phones. The apps are available through native distribution platforms, called as app stores, which are operated by the owners of the mobile operating system. Some of the most popular operating system-native stores are Apple's App Store, Google Play, as well as Windows Phone Store and BlackBerry App World. As per the 2015 report, Google play contains 2,200,000 apps, apple app store contains 2,000,000 apps and windows store has 669000. This huge number of app count is very tough to categorize, compare and suggest. To incorporate this huge size app list, data mining techniques are used.

In specific, the task of calculating relationship between apps is more valuable and not easy at all. There is a need to group related apps by its features and similarities. Using this task, app retrieval and app recommendation are easily performed. The other part of this paper describes the detection of malicious apps according to its features and permission required by the apps.

When discussing the malicious app detection process, we need to know the purpose and impact after spread the app.If the attacker spreads malicious app, the malicious can reach large numbers of users and their friends to spread spam. The main problem of malicious app is that can obtain users personal information from their mobile such as e-mail address, location, pictures and other details. So, detection of malicious apps along with the similarity detection is also an important task. This paper aims to detect malicious as well as most similar apps from the Google play store.

II. PROBLEM DEFINITION

Finding App relationship and similarity is an iterative process, which meets two imperfections. One is that it needs to run once more when novel apps appear. Hence, it is time consuming. The other is that this iterative process needs to set two initial factors. Certain results denoted that initial parameters deeply affect calculating the outcomes. But, it is difficult to determine which factor is suitable to determine the App similarity and relationship calculation.

The research community has paid little attention to Google play store, Apple store apps specifically. Most research related to duplicates, spam and malware on social media has focused on detecting malicious contents and social spam postings. Google play store has dismantled its app rating functionality recently. A recent work studies how app permissions and community ratings correlate to privacy and security risks of Google play store apps. At last, there are several complications misleads the App usage in the real-time apps on social Medias. So detection and suggestion of valuable apps with the elimination of duplicate and harmful featured apps is important.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 9, September 2016

III. RELATED WORK

At the present time, people are using smart phones for different purposes, so different and numerous mobile apps are available in the internet. Handling the large amount of apps in the internet, the task of how to define their similarities, relationship becomes more and more useful. By performing the similarity calculation, app retrieval and app recommendation are easy to be performed.

In Kim, Jognwoo, et al proposed a personalized recommendation system for mobile application software (app) to mobile user using semantic relations of apps. To do that, the authors define semantic relations between apps consumed by a specific member and his/her social members using Ontology framework. With the help of app associations, this identifies the most similar social members from the analysis method. The analysis is discovered from measuring the common features between apps used by the specific set of members. The more features shared by the users, the more similar is their preference for consuming apps. This also developed a prototype of the system using OWL (Ontology Web Language) by defining ontology-based semantic relations among 50 mobile apps in the internet. With the proposed prototype, the authors demonstrated the feasibility of the recommendation algorithm.

In Chen, Lei, Chong Wu, and Yilan Dai proposed an iterative process for app relationship detection. It combines the review similarity and app similarity together. So the authors used the iterative process to dig deep relationship among apps via reviews and feedbacks. It finds the given two apps are similar in their objective or not. This iterative process has two ways to run and only needs to set one initial parameter. By this iterative procedure, deep relationship among apps and topic similarity among reviews can be both attained. But, there are many kinds of relationship among apps which are very tough to identify. So the authors have concluded, that the iterative process can only detect there is relationship and gives its value, however, cannot tell which type of relationship it belongs to. The authors left some app relationship classification process as future process.

In [7] Kim, Junhyoung, Tae Guen Kim, and Eul Gyu Im proposed a method for Android malware similarity and clustering process. The structural information that is extracted from methods in given applications is compared to match the similar apps in the targeted application with various factors, and the number of matched methods and the total number of methods are used for similarity calculation. All the structured information's are used for the comparison process. This used DBSCAN algorithm for clustering. It also suggested clustering mechanism that can provide some feedbacks about the malware apps to others.

As a summarized view, the usual way to calculate app relationship and similarities to extract attributes from app's description and other sources to represent apps similarity measurements has several challenging tasks. From the literature, we can summarize that there only few researches made for App similarity and malicious detection problems using data mining. The above methods have the following limitations.

- One is that it can only detect shallow similarity between apps. So it doesn't give deep variations
- Some useful information is unaware and insensible, such as the viewpoint in user's review.

For this reason, this paper proposes a novel approach to combine app relationship calculation and review similarity calculation together for duplicate app and malicious app detection. The proposed work can calculate app similarity accurately. It can find more general relationships between apps, which are collected from Google play store. Additionally, several improvements are made on this proposal. One is obtaining high-quality results and the other is to replace calculation by matrix product to reduce time.And finally the malicious and similar apps are filtered at the time of recommendation.

IV. PROPOSED SYSTEM

The paper proposes a new framework is named as developed **MAP (Malicious App Prediction),** an accurate classifier for detecting malicious, and duplicate applications from Google play store. The MAP framework consist of different algorithms such as **(i) Context Based Rule Mining Algorithm (ii) Root finding algorithm (iii) Semi supervised feature Classification algorithm.** This helps to classify the mobile apps based on its similar feature and it reduces the training process in the classification. It's a self learning paradigm.

The proposed work classifies different apps based on its features and security levels. The recent methods are conducted mostly relying on user's log or app's details, which can only detect whether two apps are downloaded or used by the user. Hence, apps contain many general relationships other than similarity, such as one app needs many permissions, the proposed work classifies the app based on the permission asked by the app. These relationships cannot be performed without the whole details of app descriptions. Reviews contain user's viewpoint and judgment to apps, thus they can be used to calculate relationship between apps. To use reviews, this paper proposes a similarity and rule matching process by combining review similarity and app rule verification.

The fig 1.0 shows the overall process performed by MAP framework. Initially the App details are given as the input, after uploading the app details the relevant features are extracted and effective features are stored in the feature data base. The next step is the process of association finding, where the selected app features are crawled and that will be used to the similarity calculation.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 9, September 2016

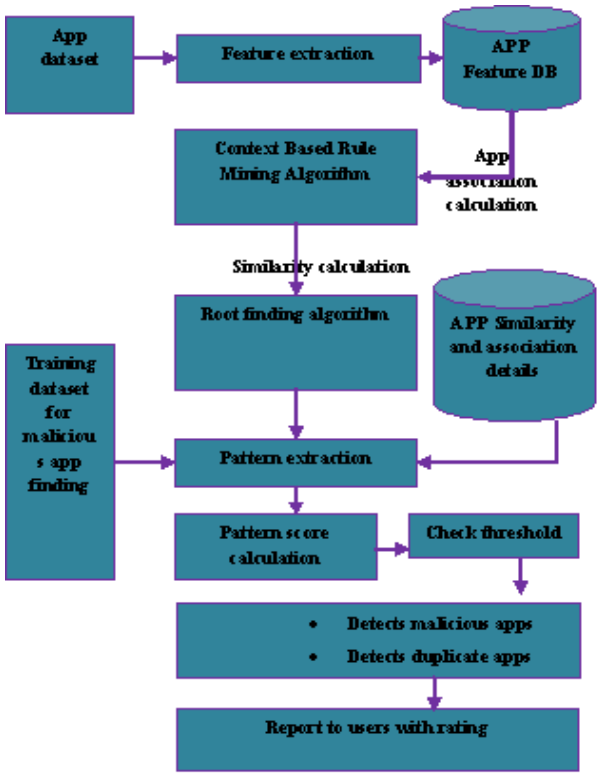

Fig 1.0 MAP framework

For association calculation, Context based rule mining algorithm is used. To retrieve apps effectively it is necessary that app should be properly classified. In order to classify the apps properly and effectively, there is need to select effective classification algorithm. There are many classification methods are in used in the literature for feature based classification such as Decision tree categorization, Rule based categorization, neural networks, Support vector machine, Bayesian categorization and many. From the above the rule based categorization is performed for effective app classification. The MAP framework consists of the following data mining algorithms for the detection and suggestion of app feature.

a. Context Based Rule Mining Algorithm:

Context Based Rule Mining Algorithmis a form of association rule to find the association between different apps. Context Based Rules claims more accuracy in association rule mining by considering a hidden variable named context variable, which changes the final set of association rules depending upon the value of context variables extracted from the mobile apps.

Step 1: scan data base D
Step 2: From D, extract each feature F and attributes A. do
Fs=extract (F (D))
Step 3: after step 2, do support calculation

$$\textbf{Support (X) = (T(x))/n} \qquad \textbf{(1)}$$

In equation (1), x is an attribute, T(x) is the total number of occurrences of x and n is the total transaction in the dataset. In the proposed system the app data's are collected and applied for frequency detection.
Step 4: Confidence calculation

$$\textbf{Confidence (X}\rightarrow\textbf{Y) =P(Y/X) (2)}$$

Where x is an attribute/ feature, Y is the total number of occurrences of Y and X is the total number of occurrences of X is the total transaction in the dataset. In the proposed system the app data's are collected and applied for rule mining. The rule generation from the confidence is important to analyze the association between attributes.
Step 5: Rule detection based on the feature and its support
The above algorithm have 5 steps totally, this helps to find the important features and relationship among different apps based on the features.

b. Root finding algorithm:

The Root finding algorithm is used to categorize and finding the app similarity. The similarity calculation process helps to know the given two apps are similar by its features and dissimilar. The straight forward way to calculate the relationship between two apps (e.g. appp and appq) is touse their app vectors as bases, such as
Calculates the similarity Sim(App1, App2)=$\sum_{F=0}^{n}\binom{n}{F}A^{F}B^{n-A}$

Here F is the feature and n is the total number of features extracted. And the A is the app1 and B is represented as app2. From the features of A, the B's features are matched and calculated the similarity between A and B. the use of Root finding algorithm is listed below.

- Based on the association rule, the similarity will be detected.
- It also detects the category and sub category of apps.

c. Semi supervised feature Classification algorithm:

The third step is the process of classification based on the similarity and features. This helps to classify the mobile apps based on its similar feature. And it reduces the training process in the classification. It's a self learning paradigm. And with the use of semi supervised learning process, the malicious app can also detect. This paper brings a brief content of malicious app detection.This includes the following process. Training using the section a and b. and finally classification of apps is performed. As like similarity measures, the malicious activities of app also can be detected. For all this the MAP framework trains set of malicious rules to the database. MAP classifier on the entire Ddataset and use this classifier to identify new maliciousapps.

V. IMPLEMENTATION AND RESULTS

The Experimental analysis is indented to be of use to show the results and findings of the research work. It has two goals: first, to provide a useful guide to new experiment lists about how such work can best be performed and written up and the next one is to challenge current

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 9, September 2016

researchers to think about whether their own work might be improved from a scientific point of view. Efficient implementations allow one to perform experiments on more and/or larger instances or to finish the study more quickly.

The proposed MAP frameworkis implemented using .Net framework. The system has taken n number of apps from Google play store. The followings are the features extracted from the mobile app.

Table 1.0 Sample features and attributes of mobile app

App name and provider	Updated date	size	installs	Current-version	Content rating	Interactive elements	permissions

The Table 1.0 shows the extracted attributes and features of a mobile app. For the implementation, every app details are extracted and converted into the proper dataset. While retrieving the features of the app, the lists of permissions are also extracted. The permission list is a general format of terms and conditions, which needs the customer's permission to access certain contents. Sometimes the permission lists are huge and unrelated to the app. In such case the malicious behavior can be predicted.

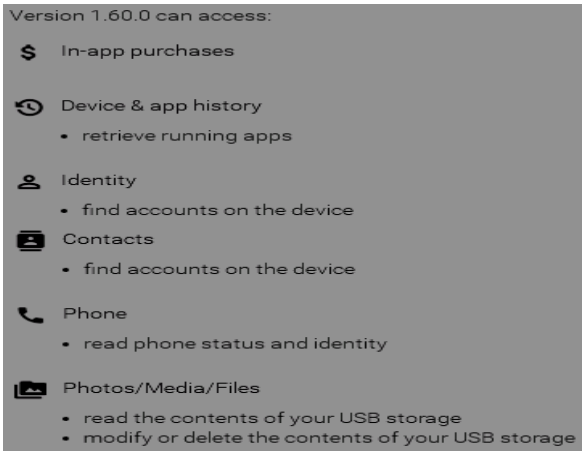

Fig 2.0 sample permission list of an APP

The Fig 2.0 list out the access permission of an application extracted from Google play store.This shows the permission groups an app will be able to access. This information can help tothe user to decide whether want to install the app or not.Extracted Apps from Google Play must also follow rules from the rule set. It removes apps that are found to violate these policies and finds the malicious one. It also has systems that analyze apps, along with various information's to protect user'sdevice against potentially harmful apps.The first set of experiments is to compare the performance of different combinations of app test, similarity and relationship based strategies with and without using MAP.

All apps are compared with one another. There are totally 60 apps are retrieved and compared. In more specific this paper particularly interested to find the time taken to find rules for the total number of features and attributes. And malicious prediction from the set of permissions during app installation and the accuracy of the detection are summarized.

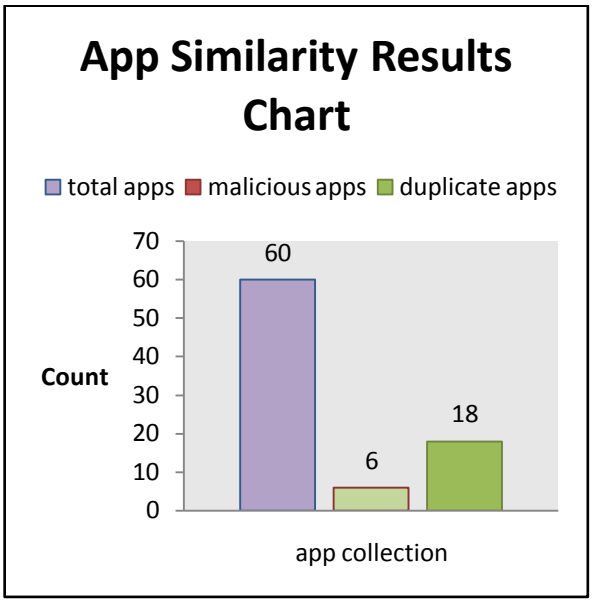

Fig 3.0 App comparison chart

After implemented the MAP, the system detected 6 malicious apps and 18 duplicate apps form the total 60 apps. The MAP framework is effective in rule creation and detection accuracy.

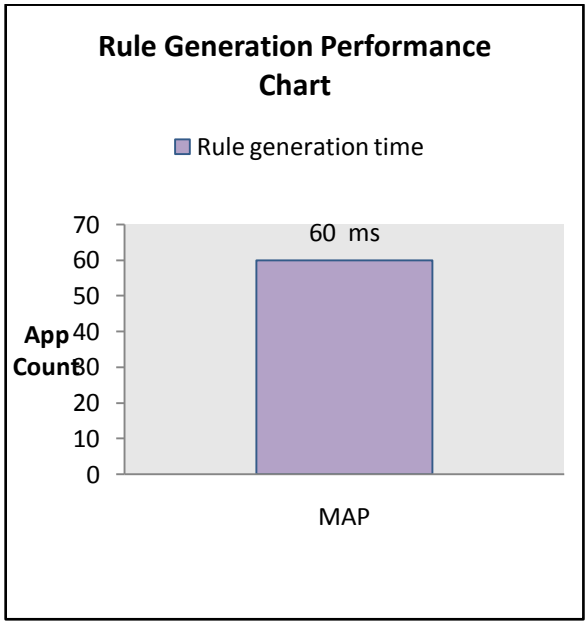

Fig 4.0 (a) Rule generation performance chart

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 9, September 2016

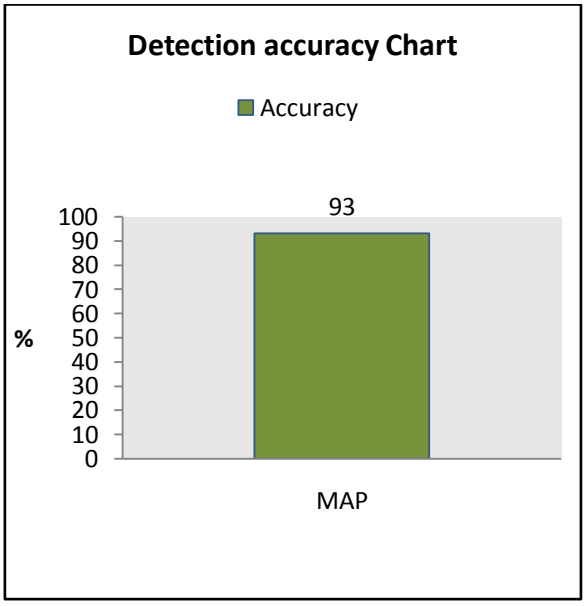

Fig 4.0 (b) detection accuracy chart

The fig 4.0 (a) and (b) shows the performance results of the proposed MAP framework. The rule generation chart shows the total time taken for creating the rule from the app dataset. And the proposed system takes 60 ms for rule generation. And the detection accuracy is measured for the above data and the total accuracy is 93%. This process shows the proposed framework gained better result.

VI. CONCLUSION

This paper proposes a new framework to calculate app similarity between apps. It combines app relationship and feature similarity and rule matching process as an iterative calculating process and finds the duplicate and malicious apps. MAP has two ways to run and only needs to set one initial parameter and malicious detection rules. By this process, the relationship between apps can be calculated accurately. The MAP covered by several algorithms for different process. Finally finds the duplicate and malicious apps among the given dataset. And it performs the suggestion process to the users based on the features and reviews. This paper also makes two improvements on existing app relationship calculation process. One is to make it high-quality even with weak initial parameters and minimum rules. The other is to reduce rule generation time and increases the accuracy.

REFERENCES

[1] Liu, Ming, et al. "APP Relationship Calculation: An Iterative Process." IEEE Transactions on Knowledge and Data Engineering 27.8 (2015): 2049-2063.
[2] https://www.statista.com/topics/1002/mobile-app-usage/
[3] Rahman, Sazzadur, et al. "Detecting Malicious Facebook Applications."IEEE/ACM Transactions on Networking 24.2 (2016): 773-787.
[4] P. Chia, Y. Yamamoto, and N. Asokan, “Is this app safe? A large scale study on application permissions and risk signals,” in Proc. WWW, 2012, pp. 311–320.
[5] Kim, Jognwoo, et al. "Recommendation algorithm of the app store by using semantic relations between apps." The Journal of Supercomputing 65.1 (2013): 16-26.
[6] Chen, Lei, Chong Wu, and Yilan Dai. "Find relationship among applications."International Journal of Networking and Virtual Organisations 15.1 (2015): 80-98.
[7] Kim, Junhyoung, Tae Guen Kim, and EulGyuIm. "Structural information based malicious app similarity calculation and clustering." Proceedings of the 2015 Conference on research in adaptive and convergent systems. ACM, 2015.