Middle-East Journal of Scientific Research 25 (4): 776-781, 2017
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2017.776.781

Amobile App Permission Recommendation for Naïve Users by Using Machine Learning Approaches

[1]M. Thenmozhi and [2]K. Nagesh

[1]Assistant Professor, Department of Computer Science,
Pondicherry Engineering College, Puducherry, India
[2]M.Tech (DCS), Department of Computer Science,
Pondicherry Engineering College, Puducherry, India

Abstract: The usage of smart phones has been tremendously increasing for the past few years. The main reason behind this growth is the adoption of app store which is offered by Apple, Google etc. Mobile app ecosystems such as Android or iOS provide standard APIs through which the mobile applications can be easily developed and distributed through app stores. These mobile applications have features like location access, contact access etc., which may access a user's private information. Very few users have experience in using the mobile application features. Other inexperienced users may allow access for all the features while installing a mobile application. This may pose several security and privacy issues. Hence, in this paper we focus on recommending mobile application features by applying different machine learning techniques. This can significantly reduce user load while allowing users to better control their mobile application features based on like-minded group preferences. From our experimental results it has been observed that the proposed recommendation system can provide accurate privacy decision to the naive user.

Key words: Security · Single Value Decomposition · Hierarchical Clustering · Random Forest Classifier · Naive

INTRODUCTION

As Android was introduced in the year 2008 it has gained a tremendous number of users over last few years. Smartphones are one of the fastest growing technology. Mobile applications are essential to the smartphone experience. The market offers a wide variety of applications like entertainment, business management, productivity, healthcare, home security etc. Many users depend upon mobile devices and applications. Inexperienced users do not know whether an application is safe to use or not and they urge to use an application [1]. As mobile application gaining its popularity, the privacy and security become a concern. The third party malicious applications steal private information of the users like contacts, messages, location etc through app permissions that lead to loss of user data and also cause financial loss. It is hard to verify the applications are legitimate or not in app market places. Unlike iOS, many android users root their devices to install apps from ”unknown sources”. This gives to install pirated, corrupted or banned apps from Google Play store by changing a systems setting and exposes their privacy to significant security risks. Some of the security issues of android phones are information leakage, permission escalation, repackaging apps to inject malicious code, colluding and Denial of Service (DoS) attacks [6].

Every trusted mobile application access some features of the mobile device which are relevant when needed. But some untrusted malicious application access the features of the mobile device which are irrelevant and causes harm. In the existing works, the recommendation system help inexperienced users to guide and use the secured features of the mobile application. But these recommendation system do not help the users to take right security decision and may not be accurate in all scenarios.

Corresponding Author: Thenmozhi. M, Assistant Professor, Department of Computer Science,
Pondicherry Engineering College, Puducherry, India.

This paper focus on recommending app permission by applying different machine learning techniques. This can significantly reduce user load while allowing users to better control their mobile app features based on like-minded group preferences. From the existing user preferences, we obtain the preference matrix. Then we obtain the feature vectors like userid, appid etc..using single value decomposition technique over the preference matrix. Then we generate the like-minded groups by applying hierarchical clustering algorithm using the features obtained. Finally, using random forest classifier we classify group users and provide accurate privacy decision to the naive user.

Related Work: In existing approach, they evaluate whether Android users pay attention to, understand and act on permission information during installation. So they have present recommendations for improving user attention and comprehension, as well as identify open challenges. Some of the techniques they have used are short-term recommendation system, low-RiskWarnings, Absent Permissions, Optional Permissions [2]. In investigating Ad Fraud in android applications they have identified two fraudulent ad behaviors in apps: 1) requesting ads while the application is running in the background and 2) clicking Without user interaction. MAdFraud tool is used which automatically runs on many apps simultaneously in emulators which trigger and expose the ad fraud [5]. The emerging applications such as Near Field Communication application that requires new levels of security that cannot be enforced by current smartphone operating systems. A generic OS framework that facilitates the creation of secure smartphone systems has been developed. The framework is used to solve four issues in smartphone security: Delayed system updates, Linux kernel, rooted phones android permission systems [3]. ProtectMyPrivacy (PMP), a system for iOS devices to detect access to private data and protect users by substituting anonymized data in its place if users decide. They have used a novel crowdsourced recommendation engine driven by users who contribute their protection decisions, which provides app specific privacy recommendations. They show the effectiveness of users privacy decision, users recommendation thereby helping to make privacy decisions [8]. A crowdsourcing recommendation framework called RecDroid [4] that facilitates a user-help-user environment regarding mobile app permission control. In this framework, the responses from expert users are aggregated and recommended to the inexperienced users. RecDroid is used to assist inexperienced users to make a right permission granting decisions [1].

Proposed System Design: In this proposed work we perform analysis of the existing user privacy preferences for granting permissions to different mobile apps. The proposed system acts as a recommendation app for the naive user. The users before installing any new mobile app can enter the name of the app in our recommendation app, which then sends the request to the recommendation system running in a server. Based on the response received from the recommendation system the recommendation app displays the app permission that can be safely selected by the users while installation. Fig 1 represents the overall architecture of the system. It shows how the client and server send the request and responds for accessing the application for privacy decision. Fig 2 represents the overview of the proposed recommendation system. The proposed work adopts collaborative filtering for building the recommendation system. First, we generate preference matrix containing existing user decisions (allow / deny) over the app permissions from the data set. We define a preference matrix #User × #app_permission matrix of M preferences, where each value in the matrix corresponds to a user's decision for a given app-permission [7]. Specifically:

$$M[u][m] = \begin{cases} +1, \text{if user u chose "Accept" for app\_permission m} \\ -1, \text{if user u chose "Reject" for app\_permission m} \\ 0, \text{if no selection} \end{cases} \quad (1)$$

By applying Singular Value Decomposition over the preference matrix we obtain the feature vectors (e.g. User ids, App IDs & Permission IDs etc.). Next, we generate like-minded groups by applying hierarchical clustering algorithm using the features obtained. Finally, using random forest classifier we classify group users and provide accurate privacy decision to the naive user.

Data Pre-Processing: In order to over the issues such as size of the dataset processed and sparsity of the preference matrix the first step of the proposed work is to apply matrix factorization.. Singular value decomposition is one of the matrix factorization technique (SVD). SVD is used to produce a limited number of eigenvectors that collectively capture most of the information contained in the original dataset. It is computationally intractable for very large databases and useful for dimensionality

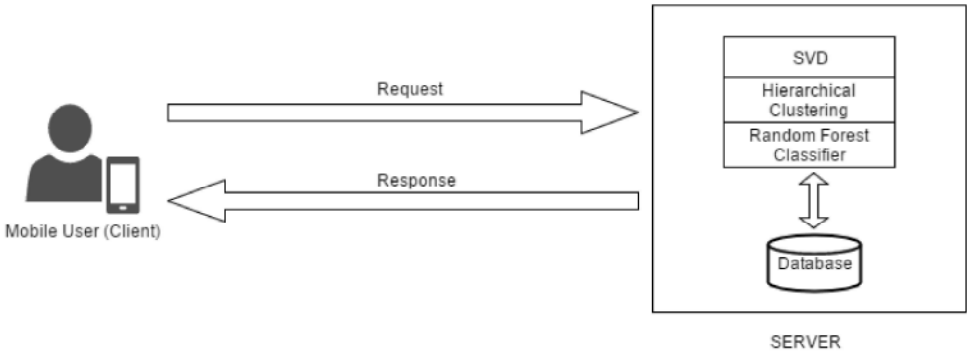

Fig. 1: Architecture Diagram

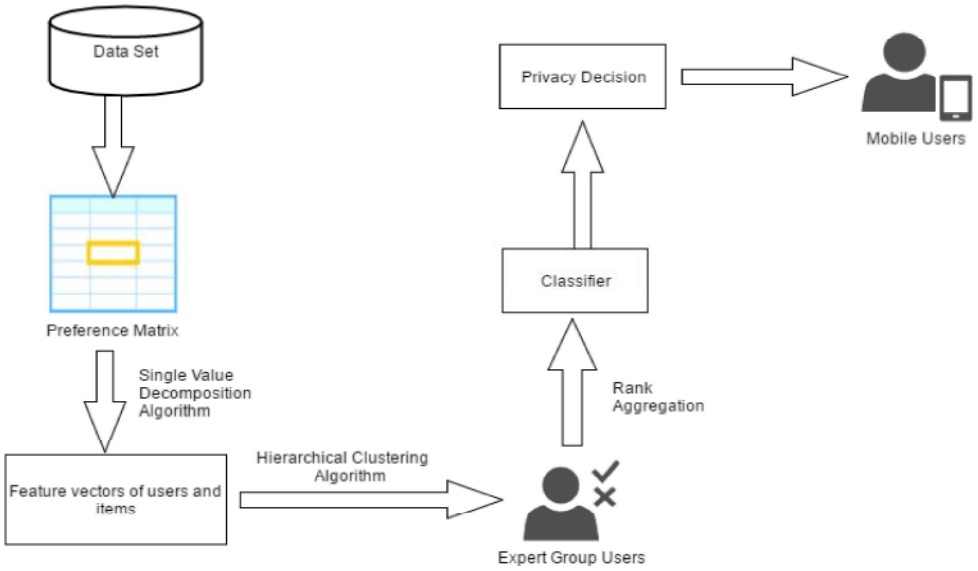

Fig. 2: Overview of proposed work

reduction of large data set. It takes a rectangular matrix which is defined as A, where A is a n x f matrix in which the n rows represents the users and f columns represents the app features. The SVD theorem states:

$$A_{nxf} = U_{nxn}\, S_{nxf}\, V^{T}_{fxf} \quad (2)$$

where
$U^{T}U = I_{nxf}$
$V^{T}V = I_{fxf}$ (i.e. U and V are orthogonal)

Where the columns of U are the left singular vectors; S has singular values and is diagonal; and V^{T} are the right singular vectors [11]. By applying single value decomposition technique over the preference matrix we obtain the feature vectors like userid, appid etc. Fig 3 represents single value decomposition technique applied to large data set to obtain the preference matrix. In this figure A_nP_n is the application permission id, AF_n are the features of the application. Thus, by applying SVD we obtain a compact matrix representing the original dataset by projecting the data along limited eigenvectors.

Data Analysis

Identifying Like Minded Users: Each user can be modeled as vector or app-permission decisions. Aggregation of user preferences along the mobile app permissions help to identify user clusters who depict common preferences. Clustering is nothing but the grouping of data objects. Here we use Agglomerative Hierarchical clustering algorithm. It is used to group the data one by one on basis of nearest distance measure of all the pairwise distance between the data point (users). The distance between the data point is recalculated and grouped until a cluster is formed. By applying this algorithm, we can obtain different like-minded users.

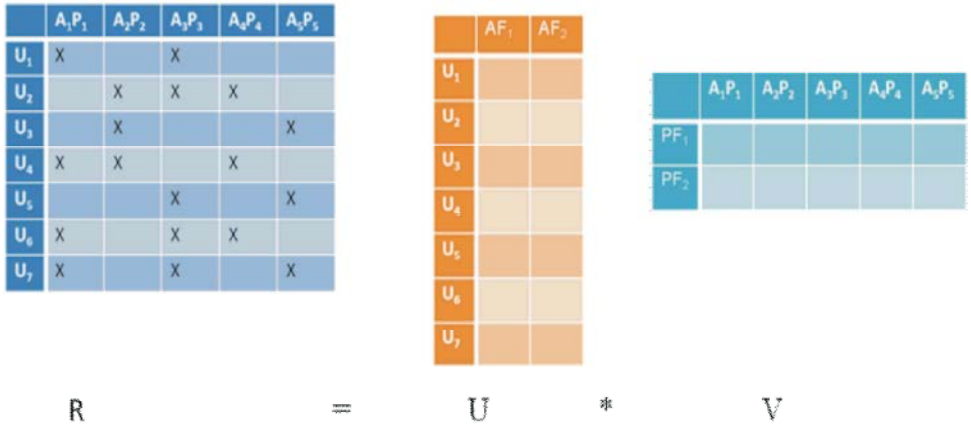

Fig. 3: Singular value decomposition technique

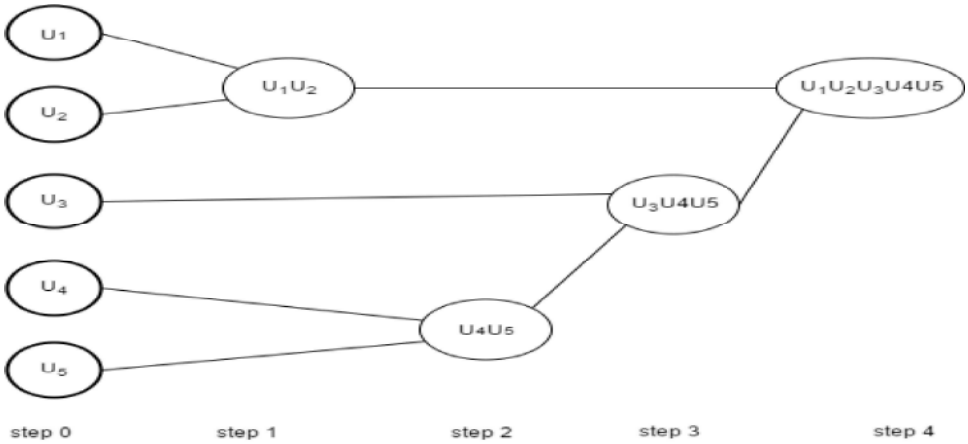

Fig. 4: Cluster Formation

Algorithmic steps for Agglomerative Hierarchical clustering

Given:

A set U of users $\{u_1,u_2,...u_n\}$

Distance function dist (a_1,a_2)

for i=1 to n

$a_i=\{u_i\}$

end for

$A = \{a_1,.....,a_n\}$

l=n+1

while A.size > 1 do

(a_{min1},a_{min2}) = minimum dist(a_i,a_j) for all a_i,a_j in A

remove a_i and a_j from A

add $\{ a_{min1},a_{min2}\}$ to A

l = l+1

end while

In the above agglomerative hierarchical clustering algorithm shows how the distance between the data points (users) have been calculated and like-minded clusters of users are formed. Figure 4 shows the cluster formation of like-minded users.

Predicting User's App Permission Preference: The next step is to build a classifier that could be used to predict a user's mobile app permission settings. A classifier is used to classify the given set of categories from the given set of data. For building the classifier we have used random forest classifier technique which is used to classify group users and provide accurate privacy decision to the naive user. Random forest classifier is a versatile machine learning method which is capable of performing both regression and classification tasks. It is a type of ensemble learning method, where a group of weak models combines to form a powerful model. It uses bagging technique for building an ensemble of decision trees [10]. Figure 5 shows the random forest classifier how different set of like-minded users for different applications are classified into various categories.

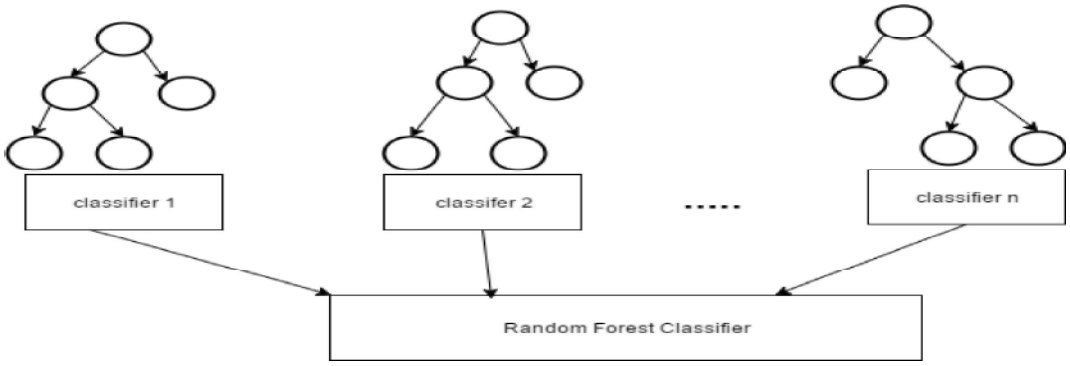

Fig. 5: Random Forest Classifier

Algorithm: Random Forest
Precondition: A training set $C := (x_1, y_1),....., (x_n, y_n)$, features F and number of trees in forest N.
function RandomForest(C, F)
$H \leftarrow \square$
for $i \in 1,...., N$ **do**
$C^{(i)} \leftarrow$ A bootstrap sample from C
$h_i \leftarrow$ RandomizedTreeLearn($C^{(i)}$, F)
$H \leftarrow H \square \{h_i\}$
end for
return H
end function
function RandomizedTreeLearn(C, F)
At each node:
$f \leftarrow$ very small subset of F
Split on best feature in f
return The learned tree
end function

The above classification algorithm is executed for each cluster obtained in the previous step. Based on the classifier built, it can be further used for prediction privacy preferences.

Naive User Recommendation: In order to predict the privacy decision for a given (naive) user for a new mobile app, his membership value with respect to each cluster is computed. For computing the membership, the given user's previous preference i.e, other app permissions chosen by the user are compared with the privacy preferences of users in individual clusters. When a particular cluster has more privacy preferences matching, then the user belong to this cluster group. Based on the group references, the classifier is used to predict the privacy/ permission decisions for the new mobile app to be installed by the user. The membership of i^{th} user in the k^{th} cluster [12] is calculated as follows:

$$\text{memb}_k^i = \frac{\sum_{i=1}^{|c_k|} u_{il}}{\sum_{c_j \in \{1 \ldots c_p\}} \sum_{1}^{|c_j|} u_{il}} \quad (3)$$

where, C_1 to C_p are all clusters that contain the i^{th} user. U_{il} calculate the degree of interest similarity between i^{th} user and the l^{th} user. Finally the top-N number of recommendation list of items is extracted from the clusters the user exists and recommended to the user.

RESULTS AND DISCUSSION

We evaluate and compare the performance of the recommendation system using coverage and accuracy metrics with the existing system. We define coverage as the number of recommendation from the given set of users.

$$C(M) = \frac{Total\ number\ of\ recommendation}{Total\ number\ of\ users} \quad (4)$$

Accuracy measures the ratio of correct predictions to the total number of users evaluated. The accuracy can be calculated by below formula:

$$A(M) = \frac{Total\ number\ of\ correct\ recommendation}{Total\ number\ of\ users} \quad (5)$$

The experiments are conducted with the user preference dataset to check the performance of the existing system [1] and proposed system. In the existing system and proposed system we evaluate the coverage and accuracy for the given set of 3793 users.

Table 1: Evaluation Metrics

	Existing System (For 3793 Users)	Proposed System (For 3793 Users)
Coverage	80	90
Accuracy	65	76

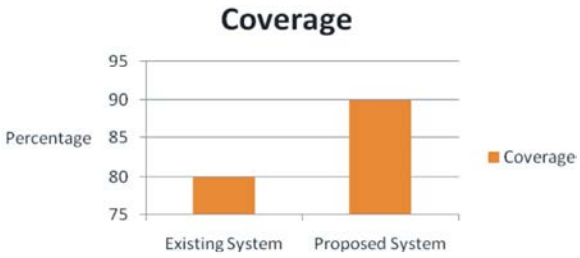

Fig. 6: Coverage

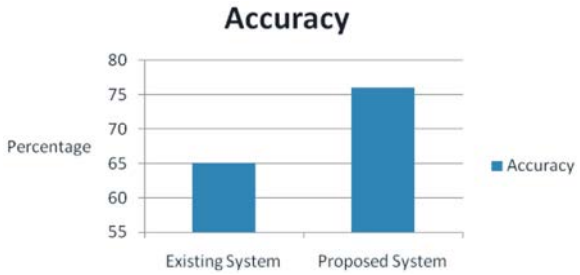

Fig. 7: Accuracy

The above table and figures shows the results obtained by existing and proposed system from the users preference data set. As shown in graph, proposed system performs better than the existing system for the users preference dataset, which indicates that improvement in accuracy and coverage for better recommendation.

CONCLUSION

The objective of the proposed work is to provide personalized privacy assistance to naïve mobile app users. Though existing works focus not providing inexperienced user with privacy decisions the results obtained were not promising. Hence, in the proposed work by applying different machine learning techniques it is possible to recommend inexperienced user with app permission preference which can help users to better control their mobile app permissions. Our experimental result shows that the proposed recommendation system achieves high accuracy and coverage to provide better recommended system to the naive users.

REFERENCES

1. Bahman Rashidi, Carol Fung, Tam Vu android fine-grained permission control system with real-time expert recommendations, Pervasive and Mobile Computing, pp: 1574-1192, 2016.
2. Felt, A.P., E. Ha, S. Egelman, A. Haney and E. Chin, 2012. D. Wagner android permissions: User attention, comprehension and behavior, in: SOUPS'12, ACM, New York, NY, USA, pp: 1-3: 14.
3. Lange, M., S. Liebergeld, A. Lackorzynski, A. Warg and M. Peter, 2011. L4android: A generic operating system framework for secure smartphones, in: SPSMD, SPSM'11, ACM, New York, NY, USA, pp: 39-50.
4. Rashidi, B., C. Fung and T. Vu, 2014. Recdroid: A resource access permission control portal and recommendation service for smartphone users, in: Proceedings of the ACM MobiCom Workshop on Security and Privacy in Mobile Environments, SPME'14, ACM, New York, NY, USA, pp: 13-18.
5. Crussell, J., R. Stevens and H. Chen, 2014. MAdFraud: Investigating ad fraud in android applications, in: 12th CMSAS, MobiSys'14, ACM, New York, NY, USA, 2014, pp: 123-134.
6. Rashidi, Bahman and Carol Fung. A survey of android security threats and defenses." Journal of Wireless Mobile Networks, Ubiquitous Computing and Dependable Applications 6 (2015).
7. Liu, Bin, Jialiu Lin and Norman Sadeh, 2013. Reconciling Mobile App Privacy and Usability on Smartphones: Could User Privacy Profiles Help?(CMU-CS-13-128, CMU-ISR-13-114)." (2013).
8. Agarwal, Yuvraj and Malcolm Hall, 2013. ProtectMyPrivacy: detecting and mitigating privacy leaks on iOS devices using crowdsourcing. Proceeding of the 11th annual international conference on Mobile systems, applications and services. ACM, 2013.
9. Steinbach, Michael, George Karypis and Vipin Kumar, 2000. A comparison of document clustering techniques. KDD workshop on text mining, 400(1).
10. Ho, Tin Kam, 1995. Random decision forests. Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on. Vol. 1. IEEE, 1995.
11. Banerjee, Sudipto; Roy and Anindya, 2014. Linear Algebra and Matrix Analysis for Statistics, Texts in Statistical Science (1st ed.), Chapman and Hall/CRC, ISBN 978-1420095388.
12. Türksen, I. Burhan and S. Jiang, 1993. Rule base reorganization and search with a fuzzy cluster analysis. International Journal of Approximate Reasoning 9(3): 167-196.