PAPER • **OPEN ACCESS**

Feature selection for malicious android applications using Symmetrical Uncert Attribute Eval method

To cite this article: H al-kaaf *et al* 2020 *IOP Conf. Ser.: Mater. Sci. Eng.* **884** 012060

Feature selection for malicious android applications using Symmetrical Uncert Attribute Eval method

H al-kaaf[1], A Ali[2], S Shamsuddin[3] and S Hassan[4]
[1,2,3,4]School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia (UTM), Johor, Malaysia

E-mail: howida10@gmail.com

Abstract. The fast growth of tablets, smartphones has led to increase the usage of mobile applications. The Android apps have more popularity, however, the applications downloaded from third-party markets could be malware that may threaten the users' privacy. Several works used techniques to detect normal apps from malicious apps based on mining requested permissions. However, there are some set of permissions that can occur in benign and malignant applications. Redundant features could reduce the detection rate and increase the false positive rate. In this paper, we have proposed feature selection methods to identify clean and malicious applications based on selecting a set combination of permission patterns using different classification algorithms such as sequential minimal optimization (SMO), decision Tree (J48) and Naive Bayes. The experimental results show that sequential minimal optimization (SMO) combining with SymmetricalUncertAttributeEval method achieved the highest accuracy rate of 0.88, with lowest false positive rate of 0.085 and highest precision of 0.910. And the findings prove that feature selection methods enhanced the result of classification.

1. Introduction

With the development of smart mobile devices (e.g., smartphones and tablets), Android represents the most popular platform for that devices as reported by [1][2]. The trend of that availability increase the usage of apps and spreading the risk of malware since sensitive information stored on a mobile device as stated by Afonso [3] and reported by McAfee [4], there are more than 2 billion smartphones around the world, which attract many malware authors and other cyber attackers. Malicious applications also lead to unexpected behaviours such as threatening user privacy, abusing the rooting privilege and exploiting the feasibilities of device in bad action as described in [5][2]. For instance, the malware named "geimini and Droid Dream" as reported by Lu et al., [5] represents one of malwares that leads to that behaviours. In addition, the permissions employed by android operating system as mentioned by android developer [6] usually asked during the installation of an application could be lead to user data leak [7][8]. There are many significant studies have been done to detect malware using different machine learning and data mining approaches by analysing permissions such as done by authors in [9] [10] [11] [12] [13] [14].

In many applications of using machine learning, the size of a dataset is important. For example, the dataset with big size and many features will not perform well until eliminating the redundant and irrelevant attributes. There are many permissions asked by applications during the installation and some could be asked by malware and non-malware applications. Therefore, the aim of our study is to

discover the relation between malicious and the requested permission through investigating relevant features using different methods of feature selection techniques to reduce the number of the redundant and unrelated features and to reduce the running time. In addition, attribute selection approaches are used to select a small set of features that are relevant to the target concept and to reduce dimensionality of datasets with less data and enhance the performance of classification. And as a result, the visualization of the data will be easier. In our experiment, we collected 260 samples (130 benign apps and 130 malware apps) of android applications from different resources [15] [16] as described in the following sections.

2. Related work

Some studies used features selection, for instance the study done in [17] used five machine learning classifier Naïve Bayes (NB), K-nearest Neighbour (KNN), Decision Tree (J48), Multi-Layer Perceptron (MLP) and Random Forest (RF) with chi-square and information gain features selection methods to detect mobile malware based on system call features. The classifiers were evaluated using the True Positive Rate (TPR), False Positive Rate (FPR), and Accuracy. Pehlivan et al., [18] used four feature selection algorithms: CFS subset evaluator, Gain Ratio Attribute Evaluator, Relief attribute evaluator and consistency subset evaluator in addition to 5 machine learning algorithms: Bayesian classification, Regression Tree, J48 Decision Tree, Random Forest and Support Vector Machine to identify malware. The best result obtained was by combining both SVM and Relief algorithms with 50 features selected. Verma et al., [19] used the information gain algorithm of feature selection to select the best extracted features (permissions and intent-filters of the manifest files) of android application package files. The study done by Altaher et al., [20] used two features selection algorithms, Information Gain (IG) and Pearson CorrCoef (PC) to rank the individual permissions and API's calls based on their importance. Their approach achieved an accuracy of 89%. Ilham et al., [21] used filtered features methods such as Gain Ratio, Information Gain, CFS subset Evaluator and Correlation Coefficient permission-based approach to detect malwares in android applications utilizing filter feature selection algorithms to select features and machine learning algorithms Random Forest, SVM, J48 to classify applications as malware or benign. Random Forest achieved the best accurate results compared to other algorithms. Kumar et al., [22] used also feature selection approaches to distinguish between malware and benign applications based on analysing permission. This work differs from previous works by presenting SymmetricalUncertAttributeEval method to select the relevant feature to target concept. In addition, this work presents the extension of our previous study on analysing malware of android applications using machine learning algorithms.

3. Overview of the method

To overcome the problem of choosing the best feature and a suitable classifier to classify apps as malware or not malware, comparisons were made between different machine learning classifiers before applying feature selection methods and after applying feature selection approaches. The work flow of our experiment is depicted in figure 1.

3.1 Dataset Collection

Our benign samples were chosen from Google play store from (2016-2017). The malicious apps were obtained from PROGuard and Drebin dataset [15] [16]. We selected the number of normal apps to be the same as malignant apps to avoid the imbalanced dataset that can cause skewed models. We collected 260 samples (130 clean apps and 130 malware apps). The permissions of our apk files were extracted and collected using Virus Total website [23]. To conduct our experiment, we prepared four dataset with 260 instances. The first dataset contains all permissions features (45 features) without using feature selection methods, the others dataset consists of 11, 12 and 14 features respectively with 260 samples after using feature selection methods.

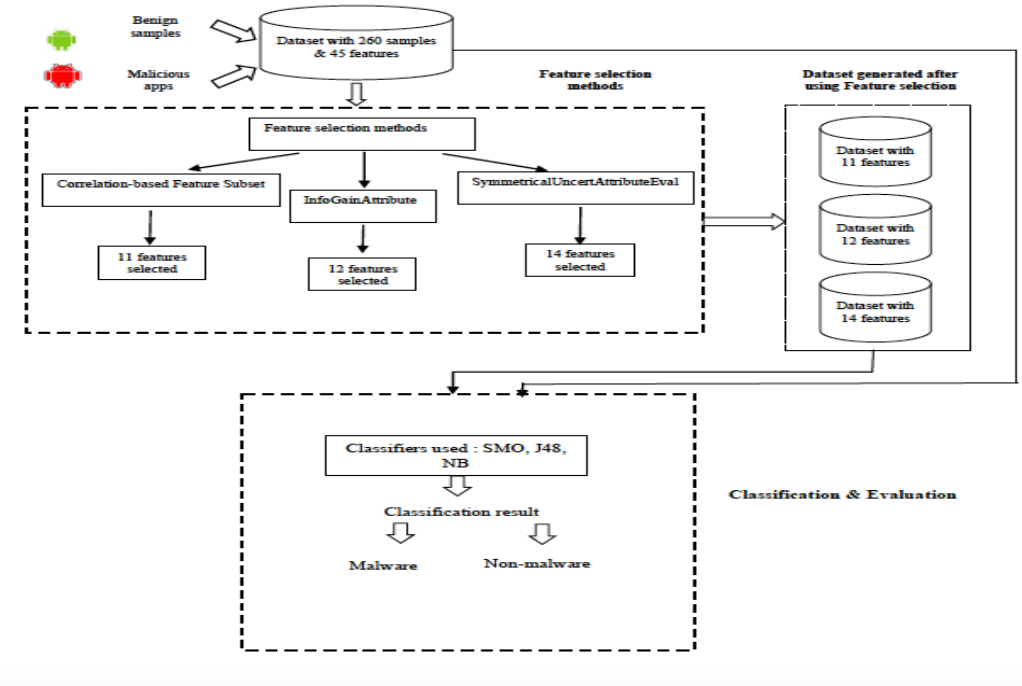

Figure 1. Workflow of the study.

3.2 Features

For this study, we used the extracted permissions as static features that were collected during the installation of apps. That features represent attributes of normal and malicious applications of our dataset. Features in our dataset are signified by a feature vector that includes all the requested permissions by apps. The presence of permission in app is denoted by 1 while the absence of the permission is represented by 0. According to android developer [6], there are more than 30 normal permissions and around 26 dangerous permissions. Therefore, we selected 45 permissions (includes all the dangerous permissions and some of the normal permissions).

3.3 Proposed Feature Selection Methods

In this paper, after generating our dataset, we used different techniques of feature selection methods to select features that are more related to class. The most important features were selected and tested based on the percentage value of the classifier accuracy. The feature selection methods used in this work were: Correlation-based Feature Subset Selection (CFS), InfoGainAttribute, and SymmetricalUncertAttribute. The explanations of feature selection approaches are displayed as follows:

1. Correlation-based Feature Subset Selection (CFS): The CFS is one method of filter algorithm that evaluates the prediction of each attribute in terms of their redundancy and the relationship between them. It selects the features that have a large correlation with the class [24] [25].
2. InfoGainAttribute: The InfoGainAttribut is one type of filter techniques that evaluates the feature according to the measurement of its information gain with respect to the class [24] [25].
3. SymmetricalUncertAttributeEval: This method evaluates the features based on the symmetrical uncertainty of each attribute. The value of the SymmetricalUncertAttributeEval is either zero or one, where one indicates that the attribute or feature is relevant to the class, while zero indicates that the attribute is irrelevant to the class [24] [25].

3.4 Classifier

In this study, we used sequential minimal optimization (SMO), Decision Tree (J48), and Naive Bayes to evaluate the feature ranking obtained during feature selection procedure. We used K-fold cross-validation to evaluate the results of choosing the best features in our dataset.

3.5 Evaluation

We used weka tool to analyse the evaluation of proposed model. We used the following metrics: Overall Accuracy, and False Positive Rate and Precision to evaluate our experiment. These measures are derived from the following basic measures explained in the following:

1. Accuracy: Accuracy is considering one of the metric used to evaluate classification models. Where TP represents the number of malware applications that classified as malware apps while FN represents the number of clean applications that incorrectly classified as malicious. TN represents the quantity of benign apps which are truly classified as benign. FN represents the quantity of abnormal apps classified incorrectly as normal [18].
2. False Positive Rate (FPR): It measures the proportion of negatives that are incorrectly identified as positive (e.g. the percentage of clean apps that misclassified as malware apps).
3. Precision: It is called also positive predicted value (PPV) which measures the proportion of positives that are considered as positive.

4. Results and discussion

This work was aimed at analysing the ability of different feature selection methods with the combination of different types of classifiers to detect malware based on permissions patterns.

4.1 Feature Subset Selection

Three dataset were created when utilizing feature selection methods to build our classification model, the lists of selected features and the feature selection approaches are displayed in table 1. For example, when using Information Gain Based Feature Selection method, 12 attributes are selected which is denoted by 12f as displayed in table 1. While using Correlation-based Feature Subset Selection (CFS), 11 features are chosen that is represented by 11f. And 14 features are selected using SymmetricalUncertAttributeEval which is represented by 14f.

Table 1. List of selected Features after using feature selection methods.

Feature selection methods	Subset of selected features
Correlation-based Feature Subset Selection CfS (11f)	(send_sms, receive_sms, read sms, read_phone_state, read_history_bookmarks, read_external_storage, wakelock, acessnetworkstate, camera, manage_accounts, use_credentials)
InfoGainAttributeEval (12f)	(send_sms, receive_sms, read sms, read_phone_state, read_history_bookmarks, read_external_storage, Get account, wake lock, access network state, camera, manage_accounts, use_credentials) .
SymmetricalUncertAttributeEval (14f)	(send_sms, receive_sms, read sms, read_phone_state, read_history_bookmarks, write_history_bookmarks, read_external_storage, Get account, wake lock, access network state, camera, manage_accounts, read_profile, use_credentials).

4.2 Classification Result

Figure1 shows the comparison of accuracy of classification algorithms before and after using feature selection methods. As we can see from table 2 & figure2, without feature selection usage, (SMO) achieved 87.6923 % accuracy rate. It is slightly same when using SMO with (CFS) and Information Gain Based feature selection method but when using SMO with SymmetricalUncert approach the accuracy rate increased to 88.4615 %. Also, using NaiveBayes with the combination of three methods of feature selection, the accuracy rate increased from 85 % to 87 %.

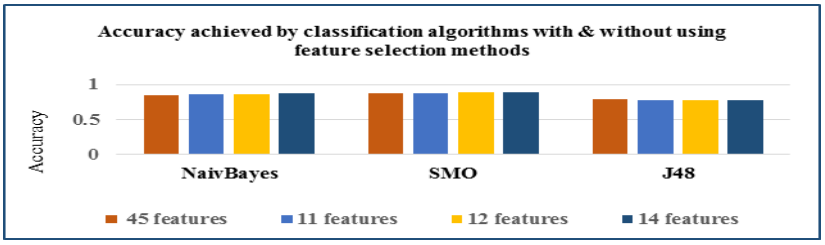

Figure 2. Achieved accuracy before and after using feature selection methods.

However, the accuracy rate is decreased when using feature selection method with J48. The accuracy rate is decreased from 79.2 % to 77.3077 % and 77.6923 % respectively after using Information Gain Based and SymmetricalUncert feature selection methods. Figure 2 shows the false positive rate (FPR) obtained by using different feature selection approaches and classification algorithms. FPR of (SMO) before applying feature selection was 0.108, but the rate is dropped to 0.085 after using feature selection methods. This means that using feature selection techniques helps in reducing FPR. FPR is decreased from 0.169 to 0.138 with NaiveBayes after using feature selection methods. FPR with J48 classifier increased from 0.185 without using feature selection to 0.246 and 0.231 respectively after using feature selection approaches which means J84 is detecting malware poorly using feature selection methods. Figure 2 clarified the FPR rate of the classifiers.

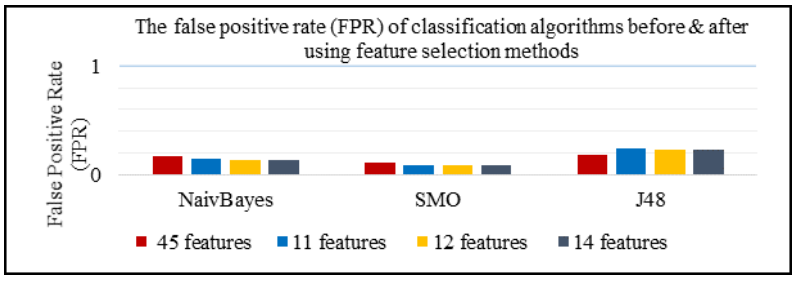

Figure 3. The false positive rate achieved by using different classification algorithms.

In terms of precision, SMO achieved the highest precision of 0.910 with the combination of SymmetricalUncertAttributeEval approach. And the rate of precision of SMO increased from 0.889 to 0.910. Also, the precision obtained by NaiveBayes is increased from 0.837 to 0.864 after using feature selection techniques. The precision obtained by using NaiveBayes, SMO, and J48 classification algorithms with & without using feature selection methods is shown in figure 3. The evaluation metrics achieved by using different feature selection and classifiers algorithms are illustrated in table 2.

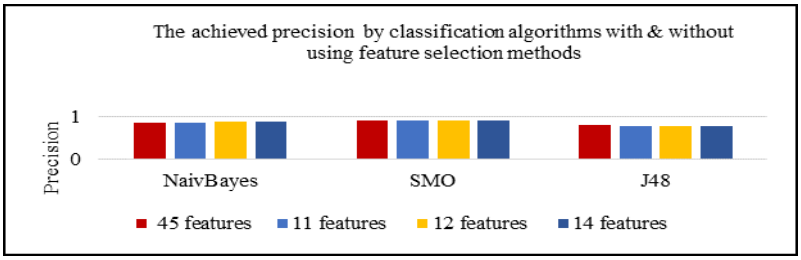

Figure 4. The precision achieved by using different types of classification algorithms.

Table 2. The evaluation metrics results of classifiers algorithms

Classifiers algorithms	Features	ACC	TPR	FPR	PREC
NaiveBayes	45f	0.85	0.869	0.169	0.837
	11f	0.857	0.862	0.146	0.855
	12f	0.865	0.869	**0.138**	0.863
	14F	**0.869**	0.877	**0.138**	0.864
SMO	45f	0.876	0.862	0.108	0.889
	11f	0.876	0.838	**0.085**	0.908
	12f	0.8807	0.846	**0.085**	0.909
	14F	**0.884**	**0.854**	**0.085**	**0.910**
	45f	0.792	0.769	0.185	0.806
J48	11f	0.773	0.792	0.246	0.763
	12f	0.776	0.785	0.231	0.773
	14F	0.780	0.792	0.231	0.774

4.3. Discussion

From our findings, we can indicate that using feature selection techniques improved the classification accuracy. SymmetricalUncertAttributeEval evaluates the worth of an attribute by measuring the symmetrical uncertainty with respect to the class [24] [25]. The selected features with that method were 14 features as displayed in table1. Our proposed (SymmetricalUncertAttributeEval) method achieved good results with using two classifiers (SMO and NaïvBayes). However, Decision Tree (J48) achieved poor accuracy result and high FPR when using feature selection techniques. That results indicate that Decision Tree (J48) combination with feature selection approach does not perform well and means J84 is detecting malware poorly. While SMO performed the best result as SMO was proved to be used in other domains such as used in classifying biomedical dataset [26].

5. Conclusion

In this paper, we study some feature selection methods to select the related permissions patterns to class target in identifying clean apps from malware apps. Our outcomes indicated that SymmetricalUncertAttributeEval technique achieved the best result with the combination of SMO algorithm with accuracy rate of 88.4615 %, with lowest FPR rate of 0.085 and highest precision of 0. 910.The obtained results show that feature selection approaches enhanced the performance of classification. For future work, we will use more collection of features and samples to get better result.

Acknowledgment
The authors would like to thank the Universiti Teknologi Malaysia (UTM) for their support in Research and Development and the Soft Computing Research Group (SCRG) for the inspiration in making this study a success. This work is supported by Ministry of Higher Education (MOHE) under Fundamental Research Grant Scheme (R. J130000.7828.4F989).

References
[1] G,Nick. Android: market share & other stats [infographic]. Retrieved from https://techjury.net/stats-about/android- market-share/, 2019.
[2] Wilkins, Z. and Zincir-Heywood, N 2019 Darwinian malware detectors: a comparison of evolutionary solutions to android malware. *Proc of the Genetic and Evolutionary Computation Conference Companion* (ACM) 1651-1658
[3] Afonso, V.M., de Amorim, M.F., Grégio, A.R.A., Junquera, G.B. and de Geus, P.L 2015 Identifying Android malware using dynamically obtained features. *Journal of Computer Virology and Hacking Techniques* **11** 9-17
[4] Macfee. Trojans, Ghosts, and More Mean Bumps Ahead for Mobile and Connected Things Trojans, Ghosts, and More Mean Bumps Ahead for Mobile and Connected Things. Retrieved from https://www.mcafee.com/enterprise/en-us/assets/reports/rp-mobile-threat-report-, 2017.
[5] Lu, X. and Huang, S.H.S 2017 Malicious Apps May Explore a Smartphone's Vulnerability to Detect One's Activities. *Int. Conf. on Advanced Information Networking and Applications* (AINA, IEEE) pp 787-794
[6] Manifest.permission. Retrieved from https://developer.android.com/reference/android/Manifest.permission.html
[7] Felt, A.P., Wang, H.J., Moshchuk, A., Hanna, S. and Chin, E 2011 Permission Re-Delegation: Attacks and Defences. *In Symp of USENIX Security* Vol 30 p 88
[8] Grace, M.C., Zhou, Y., Wang, Z. and Jiang, X 2012 *Systematic detection of capability leaks in stock android smartphones*. *In NDSS* Vol 14 p 19
[9] Peiravian, N. and Zhu, X 2013 Machine learning for android malware detection using permission and api calls. *Int. Conf. on tools with artificial intelligence* (IEEE) pp 300-305
[10] Kavitha, K., Salini, P. and Ilamathy, V 2016 Exploring the malicious android applications and reducing risk using static analysis. *Int. Conf on Electrical, Electronics, and Optimization Techniques (ICEEOT,* IEEE.) pp 1316-1319
[11] Felt, A.P., Chin, E., Hanna, S., Song, D. and Wagner, D 2011 *Proc. Int. Conf. of the 18th ACM conference on Computer and communications security* (ACM) pp 627-638
[12] Barrera, D., Kayacik, H.G., Van Oorschot, P.C. and Somayaji, A 2010 A methodology for empirical analysis of permission-based security models and its application to android. *Proc of the 17th ACM conference on Computer and communications security* (ACM) pp 73-84
[13] Zhou, Y. and Jiang, X 2012 Dissecting android malware: Characterization and evolution. *In Symp on security and privacy (*IEEE) pp 95-109
[14] Liang, S. and Du, X 2014 Permission-combination-based scheme for android mobile malware detection. *Int. Conf. on communications (ICC,* IEEE) pp 2301-2306
[15] Android PRAGuard Dataset. Retrieved from http://pralab.diee.unica.it/en/AndroidPRAGuardDataset
[16] Drebin dataset. Retrieved from https://www.sec.cs.tu-bs.de/~danarp/drebin/
[17] Mas' ud, M.Z., Sahib, S., Abdollah, M.F., Selamat, S.R. and Yusof, R 2014 Analysis of features selection and machine learning classifier in android malware detection. *Int. Conf on Information Science & Applications* (*ICISA*, IEEE) pp 1-5
[18] Pehlivan, U., Baltaci, N., Acartürk, C. and Baykal, N 2014 The analysis of feature selection methods and classification algorithms in permission based Android malware detection. *In Symp on Computational Intelligence in Cyber Security (CICS,* IEEE) pp 1-8

[19] Verma, S. and Muttoo, S.K 2016 An Android Malware Detection Framework-based on Permissions and Intents. *Defence Science Journal* **66** 618-623
[20] Altaher, A. and Barukab, O.M 2017 Intelligent Hybrid Approach for Android Malware Detection based on Permissions and API Calls. *International Journal of Advanced Computer Science and Applications* **8** 60-67
[21] Ilham, S., Abderrahim, G. and Abdelhakim, B.A 2018 Permission based malware detection in android devices. *In Proc of the 3rd International Conference on Smart City Applications* (ACM) p 83
[22] Kumar, R., Zhang, X., Khan, R.U. and Sharif, A 2019 Research on data mining of permission-induced risk for android IoT devices. *Applied Sciences* **9** 277
[23] Virus Total Malware Intelligence Services. Retrieved from https://www.virustotal.com/#/home/upload com/vtmis/
[24] Witten, I.H., Frank, E., Hall, M.A. and Pal, C.J. 2016 *Data Mining: Practical machine learning tools and techniques* (Morgan Kaufmann)
[25] Liu, H. and Motoda, H 2012 Feature selection for knowledge discovery and data mining (New York: Springer Science & Business Media) Vol 454
[26] Wosiak, A. and Dziomdziora, A 2015 Feature Selection and Classification Pairwise Combinations for High-dimensional Tumour Biomedical Datasets. *Schedae Informaticae* **24** 53