International Journal of Engineering & Technology, 7 (4.31) (2018) 209-213

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Mobile Malware Classification

Zolidah Kasiran[1]*,Norkhushaini Awang[2],Fatin Nurhanani Rusli[3]

University Teknologi MARA
40500 Shah Alam
Selangor
**Corresponding author E-mail: zolidah@tmsk.uitm.edu.my*

Abstract

Android malware is growing in such an exponential pace which lead to the need of an efficient malware intrusion detection technique. The single approach of clustering or classification technique in malware intrusion detection yield to high negative positive alarm rate.. This project had proposed clustering in intrusion detection method using hybrid learning approaches combining K-Means clustering and Naïve Bayes classification had been proposed. The result had shown the improved false rate alarm in malware detection.

Keywords: *Classification; K-means; Malware; Mobile Malware*

1. Introduction

This Mobilesmartphones is way of life in our modern community. There are few mobile platforms offered by smartphone companies such as Android, Apple iOS and BlackBerry. Android have been dominating the smartphones computing platforms over the last few years. The growth in smartphones users have encouraged in more mobile apps development, that is also attract malware developers to participate.

Android malware is growing in such an exponential pace which lead out for automated tools that can aid the malware analyst in analysing the behaviours of new malicious applications. Android provider had recommended numerous security methods to prevent malware installation especially when it involved Android permission system. Malware threats against the smartphone user are increasing especially on Android users. Due to the lack of awareness on the users, the Android malware was spreading during permission stage [1].

Android malware has become a big issue for smartphone user since Androidsmartphones had become a necessity in daily life [2]. As in Japan and Korea, smartphone function not only to call and message anymore, it is more than that. Smartphone had been used to identify users credit card and even paying bus fare was done through smartphone. However, as smartphone was used every day in every aspects, the more chances for user to exposed data lost and drawn to ransomware attack. This threat had become quite serious because it involved user's confidential data and money can be snatches away through installation of any random apps in the market.

According to [3], DroidKungfu was one of the most harmful Android malware and it has several names. For example; DroidKungfu1, DroidKungfu2, DroidKungfu3, DroidKungfu4, DroidKungfuSappand DroidKungfuLena. His research also stated the number of detection results from Kaspersky which recorded 205 detections and Dr. Web captured 310 detections. Both antivirus companies verified DroidKungfu as the highest number of the detection among other types of Android malware. The number of detections then followed by AnserverBot and BaseBridge in the second and third place.

Though there are many clustering-based method used in detecting malware, the issue of false alarm rate and accuracy is still a topic of discussion. Besides, [4] also concluded there are not even a single clustering method yield low false alarm rates with high detection rates. After a few readings, there are few papers stated that using K-Means the possibility of accuracy might be upgraded [5]. While, another paper stated that a combination of techniques may also increase the precision and reduces false alarm rates compared to stand-alone clustering or single classification [6-8].

Therefore, clustering in intrusion detection method using hybrid learning approaches combining K-Means clustering and Naïve Bayes classification had been proposed. According [9], K-Means is lightweight, easy to implement and fast-iterative algorithms compared to other clustering methods.

The remaining of this paper is organized as follows. The next section discussing the malware treats of smartphones and the malware clustering and classification methods that have been in research in recent years. Section III describes the overview for the hybrids clustering and classification methods. Section IV presents the experiment and evaluation.

The style from these instructions will adjust your fonts and line spacing. Please do not change the font sizes or line spacing to squeeze more text into a limited number of pages.

2. Related Works

An Android suffered from many security threats and malware due to the lack of efficient security tools for Android protection. The paper by [10] proposed an Android Intrusion Detection System (IDS) which presents as Mob-AIDS was developed using the Java 2 Mobile Enterprise (J2ME) platform. This method was evaluated based on user's assessment to determine its efficiency in terms of graphical user interface (GUI). Results of the analysis of the respondent's data show that Mob-AIDS has sufficient capability to prevent unauthorized or unnecessary access into the Android Mo-

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

bile enterprise. It also produced a more secured and reliable operating environment for Android.

An approach to detect an intrusion attack by clustering was proposed by [11] which used to identify groups of similar behavior object such as malicious and non-malicious activity was proposed in their research paper. Besides, classification technique using K-Means was used in [6, 12] experiments to classify all data into particular class categories. The proposed technique seems to work excellently for various types of attacks and reduce the time consuming in the malicious apps detection. In their research, as compared to only one rule classifier the clustering techniques was being used as a pre-classification component for the purpose of categorized similar data items into their respective classes which helps to produce better results.

2.1 Clustering

Clustering is one of the techniques that help to categorizing Android malware. It is one of a common data mining and statistical data analysis techniques that have been widely used by many researchers. Clustering is an unsupervised learning while classification is a supervised learning. In malware analysis research few types of clustering have been employed such as K-Means Clustering.

K-Means clustering is a popular clustering-based intrusion detection method to identify and classifies any collected dataset. This clustering type is commonly used to categorize the data into K cluster of group for example; C_1, C_2,..., C_K that represented by their means. The mean for each cluster is called “centroid” or “center. [13] had used this technique to detect malware in androids platform and reported that the techniques is possible in classifying malware. Another researcher [14] used clustering techniques and compared with Mini Batch K-means algorithm in analyzing network traffic.

2.2 Classification

Classification is a data mining function that assigned items in a collection to target categories, families and classes. This technique is a supervised learning approaches that used to classify any given dataset. Supervised learning is a condition where the classification was done according to the information from its database. A classification task begins with a dataset which the class assignment are known. There are so many classification techniques such as; Naïve Bayes, Linear Regression, Supported Vector Machine(SVM), OneR and decision tree (J48).

2.3 Naïve Bayes

Naïve Bayer’s Classification had been studied widely since the 1950s and remained a common method with appropriate preprocessing. It is competitive in this domain with more advanced methods including support vector machines. Technically, Naïve Bayes uses Bayesian Theorem, a formula that calculates a probability by counting the frequency of values and combinations of values in the historical data. It is useful in automatic medical diagnosis in biomedicine sector. Naïve Bayes classifiers are highly scalable which required a number of parameters in the number of variables in a learning problem. Furthermore, Naïve Bayes is a simple technique to produce classifiers, assign class labels to problem instances and it represented as vectors of feature values. Usually Naïve Bayes is not using single algorithm in classifiers, but a family of algorithms based on a common principle. In simple words, this classification technique assumed attributes have independent distributions. The probability can be viewed by a single scan of the database and stored them in the table.

2.4 Support Vector Machine (SVM)

Support Vector Machine (SVM) was introduced in 1963 by Vladimir N. Vapnik and Alexey Ya. Chervonenkis is a powerful algorithm based on linear and nonlinear regression. SVM classification is one of supervised learning method that focusing in decision boundaries concept. This classification techniques covered linear classifier by dividing the training data using optimal separating hyperplane.

This classification techniques able to linearly separates the dataset by applying the kernel tricks to maximum hyperplanes. The algorithm can differ with respect to accuracy, time to completion, and transparency. Researcher [15] employed SVM classification techniques vs Alligator in their projects and yield of SVM performed better. Another researchers [16] and [17] also reported the good performance result in android’s classification using SVM techniques.

3. Research Methods

This project had adopted three main phases as in Figure 1 which was Feature Extraction, Clustering & Classification. The first phase was feature extraction of the permission from the manifest file MalGenome datasets. The result from the data extraction was passed to the clustering phase adopting the K-Means clustering technique. The last phase was classification using Naïve Bayes.

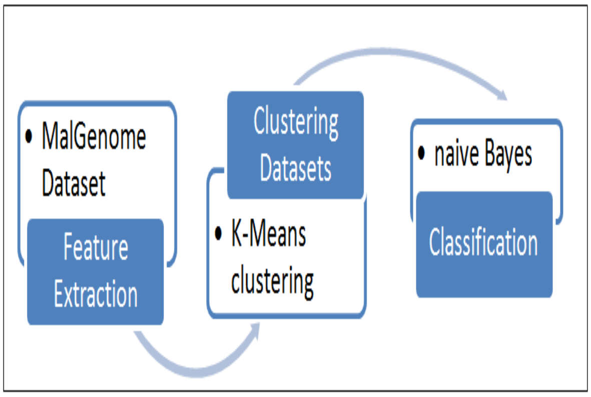

Fig 1: Reseach flows phase

In the feature extraction phase, the static analysis was done manually by converting the apk file into a normal folder that Windows can work on Malware dataset. Then, the apk files can be opened and viewed. There are a few files contains in each apk file which are “Android Manifest.xml”, “classes.dex” and yml files. From this extracting file, then the static analysis was started by observed all xml files from all different 427 apk files chosen. Subsequently, from xml files a collected of data with permission list on the apk files then accumulated in comma demilated (csv) files. The samples of Android malware dataset were in apk which only readable in Android environment. Each sample provided was in sha256 unique names.

In the manifest file, the application was asking permission on “READ_CONTACTS”, “READ_PHONE_STATE”, “RECEIVE_SMS” and “SEND_SMS”. The permissions were requested to read the contact information, write, receive and send text messages which looks suspicious.

The malware process flows for this project is shown in Fig 2. The csv file was converted into a .rff file format through. arff viewer in Weka for data extraction. Then, all experimentation was done, and K-Means clustering are used to cluster the data into clustered sets. In this hybrid learning approach, similar data was grouped based on their behaviors by applying K-Means clustering as a first classification step. Then, Naïve Bayes was used to classify the resulting clusters as a second classification step.

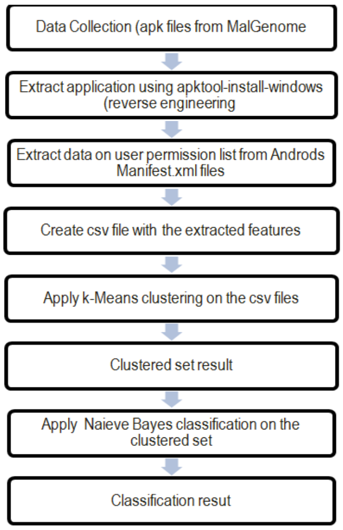
Data Collection (apk files from MalGenome
Extract application using apktool-install-windows (reverse engineering
Extract data on user permission list from Androds Manifest.xml files
Create csv file with the extracted features
Apply k-Means clustering on the csv files
Clustered set result
Apply Naieve Bayes classification on the clustered set
Classification resut

Fig. 2: Mobile Malware Classification Process Flow

The experiment was held to achieve the first and the second objective which are to cluster the Android apps into clustered set and to classify the clustered sets using Naïve Bayes classification. The clustered set then divided into cluster 0 and cluster 1. The prediction of this clustering techniques was not very precise. The single clustering methods, K-Means detects most if the malware apps as non-malware apps. Thus, the results obtained from the clustering methods then labelled into two categories where each category represent apps types; Type A (Malware) and Type B (Non-Malware). Afterward, three types of hybrid approaches were done in this experiment with the value of clusters, k is equal to 2 (k=2). There are three experiments was designed in this research project which are;

Experiment 1: Hybrid learning approaches by K-Means Clustering and Naïve Bayes Classification (KM+NB)

Experiment 2: Hybrid learning approaches by K-Means Clustering and One-R Classification (KM+1R)

Experiment 3: Hybrid learning approaches by K-Means Clustering and J48 Classification (KM+J48)

The results from each experiment was summarize in Table 1 where the performance measurement involved was True Positive Rates (TPR), False Positive Rates (FPR), Precision and Accuracy. All those performance measurements were calculated by Weka using the formula where the following term represents;

TP = No. of malware apps correctly classified as malware.

TN = No. of benign apps correctly classified as non-malware.

FP = No. of benign apps incorrectly classified as malware.

FN = No. of malware apps not detected.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$False\ Positive\ Rates\ (FPR) = \frac{FP}{FP+TN} \quad (2)$$

$$True\ Positive\ Rates\ (TPR) = \frac{TP}{TP+FN} \quad (3)$$

$$False\ Alarm = \frac{FP}{FP+TN} \quad (4)$$

Beforehand, the data collected from the AndroidManifest.xml as explained earlier need to be divided into two datasets. The first dataset was training dataset and the second one is testing dataset. As shown in Table 1, the dataset was divided into training and testing as 60:40. The dataset split does not have any significant reason, however in this research project it was due to the reference standard. It was recommended for a better result. Then both of dataset was used to implement the clustering method. Both dataset involved approximately 90 types of user's permission list.

Table 1: Sample Distribution for Training and Testing Dataset

	Malware	Non-Malware	Total
Training dataset (60%)	254	16	270
Testing dataset (40%)	173	7	180
Total	427	23	450

4. Result

All the experiments held covered all objective related to this project which are to cluster malware apps using K-Means clustering methods into clustered sets. The second objectives was to classified the clustered apps using Naive Bayes classification. K-Means clustering result with value of cluster, k=2 which cluster 0 (C0) as malware, cluster 1 (C1) as non-malware apps. After the cluster sets obtained, the data then labelled into two definite group which are malware (A) and non-malware (B). Another parameter that is commonly used is Receiver Operating Characteristics (ROC) Curve. ROC Curve is a commonly used graph that summarizes the performance of a classifier over all possible thresholds. It is generated by plotting the True Positive Rate (y-axis) against the False Positive Rate (x-axis) as the threshold adapted for assigning observations to a given class.

Experiment 1: KM+NB

Table 2 and Table 3 summarize the classification results from the experiment 1 by applying Naive Bayes classification in the K-Means clustered sets. In clustering implementation, the results obtained from K-mean was not very accurate. For training dataset, the value of Android malware apps was 254 and non-malware was 16. In this experiment, the clustered sets from K-Means clustering was used to applied Naive Bayes classification with the value of clusters, k is equal to 2 (k=2).

Table 2: Classification Result KM+NB for training dataset

Class Type	Predicted Malware	Predicted Non-malware
Malware	TN: 255	FP: 3
Non-malware	FN: 0	TP: 12

Table 3: Classification Result KM+NB for testing dataset

Class Type	Predicted Malware	Predicted Non-malware
Malware	TN: 172	FP: 1
Non-malware	FN: 0	TP:7

450 Android apps were tested for the presence of malware. Out of those 450 Android apps, it was divided 60% into training dataset and 40% testing dataset. Therefore, the dataset for training involved 270 apk files and 180 apk files was in testing dataset. In reality, 427 Android apps in the sample was a malware sample from MalGenome and 23 Android apps are non-malware sample. Table 3 shows the confusion matrix from the classification methods. For testing dataset, only one app was misclassified. According to the confusion matrix results from experiment 1, there are actually very well classified data obtained.

Experiment 2: KM+1R

Table 4: Classification Result for (KM+1R) for Testing Dataset

Class Type	Predicted Malware	Predicted Non-malware
Malware	TN: 172	FP: 1
Non-malware	FN: 0	TP: 7

There are 180 Android apps were being tested for the presence of that malware. Out of those 180 Android apps, the classifier predicted malware was 172 times, and predicted non-malware seven times for testing dataset as shown in Table 4. In reality, there are 173 Android apps in the sample was a malware sample from MalGenome and another 7 Android apps are non malware sample.

The result was very satisfied when there are only one malware apps predicted as non-malware apps. Almost all data defined was corrected. The results yield through this experiments, it shows the same value of confusion matrix with KM+NB.
Experiment 3: KM+J48

Table 5: Classification Result for (KM+J48) for Testing Dataset

Class Type	Predicted Malware	Predicted Non-malware
Malware	TN: 171	FP: 2
Non-malware	FN: 1	TP: 6

According to the confusion matrix result gather from experiment KM+J48 in Table 5 for training dataset which involves 180 malware apps and 7 non-malware apps. The results show that by applying J48 decision tree classification in cluster sets from K-Means, it detects 171 apps as the real malware. There are one missclassified in non-malware which predicted as malware. While two malware apps were predicted as non-malware. Sum of the missclassified attack was three apps in total. The accuracy, TPR and FPR then calculated and recorded in the next section.

Table 6: Results of Hybrid Learning Approaches for Training Dataset

Hybrid Method	True Positive Rate (TPR)	False Positive Rate (FPR)	F-measure	Accuracy
KM+NB	0.989	0.001	0.989	98.8
KM+1R	0.996	0.000	0.996	99.6
KM+J48	0.993	0.080	0.993	99.3

Table 7: Results of Hybrid Learning Approaches for Testing Dataset

Hybrid Method	True Positive Rate (TPR)	False Positive Rate (FPR)	F-measure	Accuracy
KM+NB	0.994	0.00	0.995	.99.4
KM+1R	0.994	0.00	0.995	99.4
KM+J48	0.983	0.138	0.984	98.3

Table 6 and Table 7 shows the results from the three experiments where the highest accuracy was from the results of experiment 1. Experiment 1 was using K-Means clustering and Naive Bayes (KM+NB) classification yield 99.4% of accuracy compared with two other hybrid methods.
The formula for False Positive Rates (FPR) was similar with False Alarm Rates. This project measures the performance of each method using accuracy, Precision, TPR and FPR only. Third objective was to evaluate the hybrid learning approaches. Table 8 show the result of Hybrid Learning Approaches from three different classification techniques with the same cluster sets from K-means clustering process.

Table 8: Results of Hybrid Learning Approaches

Measurement	Hybrid Approaches		
	KM+NB	KM+1R	KM+J48
Accuracy (%)	99.4	99.4	98.3
Precision	0.995	0.995	0.985
True Positive Rates (TPR)	0.994	0.994	0.983
False Positive Rates (FPR) (%)	0.000	0.000	0.138

The table shows the differences on accuracy between three experiments held. The first experiment (KM+NB) shows the highest accuracy compared to the other. According to [7], the combination of techniques may increase the value in term of accuracy and precision. The Decision tree (J48) classification resulted better compared to Rule-based (1R) classification [18].
Figure 3 shows the differences on accuracy between this project and previous research. The result from this research using KM+NB shows the highest accuracy compared to the previous research. The first paper by [19] show an accuracy value of 83%. His research was done on Android Malware by using dynamic analysis. This research result was similar to [8] in term of hybrid method. However, he used KDDCup 99 as his dataset which means it is a different dataset involved.

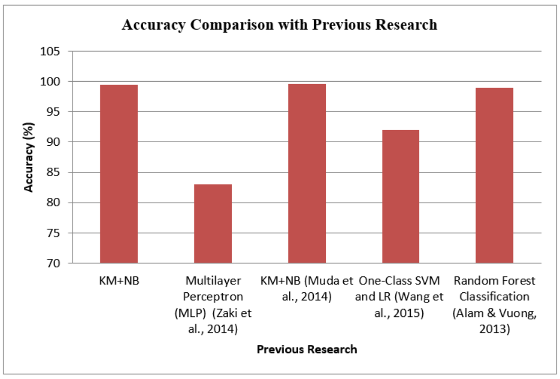

Fig 3: Results Comparison with Previous Research

4. Conclusion

After the experiment done in this research project, it can be concluded that using hybrid learning method yield better result compared to single method as mention in previous chapter. This research project did not cover any single classifier result. However, as we can see in the results from K-Means clustering the results from the clustered only had predicted so many malware apps as a normal apps. Therefore, a classification method had been proposed to overcome the flawless of the single method.
In the future, a suggestion on a combination of static and dynamic analysis in Android malware detection should be considered. Moreover, use many datasets for example MalGenome, Drebin and Andrubis may expand the results of both methods. Besides, the methods in clustering can be other than k-mean and hierarchical. Future works, another complex clustering method for example Expectation Maximization (EM) clustering and X-Means could be choose to replace K-Means clustering method for better results.

Acknowledgement

This research was financially supported by the Ministry of Higher Education through Fundamental Research Grant Scheme 600-RMI/FRGS 5/3(0006/2016).

References

[1] Wang, X., Yang, Y., & Zeng, Y. Accurate Mobile Malware Detection and Classification in the Cloud. SpringerPlus, 2015; 4(1), pp583

[2] S Anwar, M F Zolkipli , Z Inayat, J Odili , M Ali & J M Zain, Android Botnets: A Serious Threat to Android Devices, Pertanika Journal Sci. & Technology ,2018; 26 (1) pp 37 - 70

[3] Le Thanh, H." Analysis of Malware Families on Android Mobiles: Detection Characteristics Recognizable by Ordinary Phone Users and How to Fix It" . Journal of Information Security,2013; 4(October), pp213–224.

[4] Wankhade, K., Patka, S., & Thool, R. "An Efficient Approach for Intrusion Detection Using Data Mining Methods" International Conference on Advances in Computing, Communications and Informatics (ICACCI) 2013;pp 1615–1618.

[5] Sitaram, D. "Intrusion Detection System for High Volume and High Velocity Packet Streams: A Clustering Approach." International Journal of Innovation, Management and Technology,2013; 4(5).

[6] Emami, Y., Ahmadzadeh, M., Salehi, M., & Homayoun, S. "Efficient Intrusion Detection using Weighted K-means Clustering and Naïve Bayes Classification." Journal of Emerging Trends in Computing and Information Sciences, 2014;5(8),pp 620–623.

[7] Chandramohan, M., Tan, H. B. K., & Shar, L. K. “Scalable malware clustering through coarse-grained behavior modeling.” Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering,2012; pp 27.

[8] Muda, Z., Yassin, W., Sulaiman, M. N., Udzir, N. I., Technology, I., & Ehsan, S. D. “K-Means Clustering and Naive Bayes Classification for Intrusion Detection.” Journal of IT in Asia, 2014; 4(1) pp 13-25.

[9] Zhong, Y., Yamaki, H., & Takakura, H. “A grid-based clustering for low-overhead anomaly intrusion detection. “5th International Conference on Network and System Security,2011; pp 17–24.

[10] Christopher, O., Comfort, D., & James, A. (2014). An Intrusion Detection System for the Android Mobile Enterprise, IJCSI International Journal of Computer Science Issues, 2014; 11(3), pp 161–166.

[11] Singh, G., Patrick, A., & Rajpoot, L.” A Clustering based Intrusion Detection System for Storage Area Network. “International Journal of Computer Applications, 2014; 88(9), pp 14–18.

[12] Elssied, N. O. F., & Ibrahim, O.” K-Means Clustering Scheme for Enhanced Spam Detection.” Research Journal of Applied Sciences, Engineering and Technology, 2014;7(10), pp 1940–1952.

[13] Aiman A. A.S, Kangbin Y,Osama A.G. “ Analysis of Clustrering Technique in Android Malware Detection.” 7th International Conference on Innovative Mobile and Internet in Ubiquitious Computing,2013;pp 729-733

[14] Ali F, Badrul A.N, Rosli S, Fairuz, A. “Comparative Study of K-means and Mini Batch K-means Clustering Algorithms in Android Malware Detection Using Network Traffic Analysis” International Symposium on Biometrics and Security Technologies (ISBAST),2014; pp 193-197

[15] Apvrille L and Apvrille A,” Identifying Unknown Android Malware with Feature Extraction and Classification Techniques.” IEEE Trustcom/BigDataSE/ISPA, 2015; pp 182-189

[16] Wei C, David A,Andrew D.G, Charles S, Igor M. “More Semantics More Robust: Improving Android Malware classifiers, 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks”2016; pp 147-158

[17] Suyash J, Tae T,Jaehoon J,Young H K, Jeong N K. “More Semantics More Robust: Improving Android Malware Classifiers,” 31st International Conference on Advanced Information Networking and Application Workshops, 2017;pp 370-374

[18] Anuar, N. B., Sallehudin, H., Gani, A., & Zakari, O. ” Identifying False Alarm for Network Intrusion Detection System using Hybrid Data Mining and Decision Tree,”Malaysian Journal Of Computer Science, 2008; 21(2), pp 101–115.

[19] Zaki, M., Sahib, S., Abdollah, M. F., Selamat, S. R., & Yusof, R. “Analysis of Features Selection and Machine Learning Classifier in Android Malware Detection” International Conference on Information Science and Applications (ICISA), 2014; pp 1-5