International Journal of Engineering Applied Sciences and Technology, 2018
Vol. 3, Issue 3, ISSN No. 2455-2143, Pages 56-61
Published Online July 2018 in IJEAST (http://www.ijeast.com)

AN ANALYSIS OF FEATURE SELECTION METHOD IN MOBILE MALWARE DETECTION

Palak Girdhar , Deepali Virmani
Computer Science and Engineering Department
Bhagwan Parshuram Institute of Technology, Delhi

Abstract - Malware detection systems help in filtering the mobile applications and determine if it has malicious intent or not. The mobile applications used by the user are driven by a number of features. These features contain information which can detect whether an application is malignant or benign. The detection method focuses on determining the best features which can help in detecting the malware correctly. In the literature, researchers have proposed to use many feature selection methods. In this paper, a method comprising of CfsSubset Eval, Classifier SubsetEval and Principal component Analysis (methods of feature selection) have been used with Naïve Bayes classifier to analyse the effect of feature selection on the binary classification of mobile malware. The proposed system is cross validated with WEKA, a popular data mining tool. The results clearly indicate the variation in using the Naïve Bayes classifier *with and without* feature selection methods.

Keywords: Feature selection, Mobile Malware, Naïve Bayes, Filter Method, Wrapper Method

I. INTRODUCTION

With the rapid growth of smart phones in the market, there is a significant increase in the use of mobile applications. The increase poses a serious concern of security and privacy of these mobile applications. In today's era, communication is not the only use of mobile systems rather people use them for personal payments, social communications, other entertainment and many more (Shabtai, Tenenboim-Chekina, Mimran, Rokach, & Shapira, 2014). Furthermore, the significant growth of the mobile usage and easy availability of applications has caused a serious issue in mobile phone security. The intention of the attacker is to launch malware applications in the android market. These applications are governed by attributes also known as features. Features like frame number, frame length, source and destination IP, get/post methods(Narudin, Feizollah, Anuar, & Gani, 2016). These features carry information about the application as well as the developer. Careful analysis of these features can help in identifying malignant applications. (Sung & Mukkamala, 2004)

A malware detection system identifies whether an application is malignant or not. Some of the existing malware detection(Shabtai, Tenenboim-Chekina, Mimran, Rokach, & Shapira, 2014)(Narudin, Feizollah, Anuar, & Gani, 2016) (Shabtai, Kanonov, Elovici, Glezer, & Weiss, 2012) (Virmani, Taneja, Chawla, Sharma, & Kumar, 2016)(Chen, et al., 2017)(shabtai, Moskovitch, Elovici, & Chanan, 2009) (Ham & Choi, 2013)(Feizollah, Anur, Sallah, & Wahid, 2015)(Milosevic, Dehghantanha, & Raymond Choo, 2017)methods are broadly classified into two categories: *static analysis*, *dynamic system* level behavioral analysis(Chen, et al., 2017)(Coronado-De-Alba, Rodríguez-Mota, & Escamilla- Ambrosio, 2016) .

In a fine grained analysis, malware detection methods filter out malicious features. The aim of this paper is to identify the subset of the features which can effectively classify the applications into benign or infected class. The malware detection methods can use the proposed model of feature selection for accurately classifying the applications into different classes.

The rest of this paper is organized as follows: section II discusses the preliminaries used in developing the proposed model. Section III presents the importance of using feature selection methods. Section IV discusses the proposed methodology. Section V and VI gives the evaluation measures and experiments, results and analysis of the proposed method respectively. Section VII concludes the proposed work.

II. BACKGROUND

Various methods for the detection of intrusion/malware have been used in the literature(Shabtai, Kanonov, Elovici, Glezer, &

International Journal of Engineering Applied Sciences and Technology, 2018
Vol. 3, Issue 3, ISSN No. 2455-2143, Pages 56-61
Published Online July 2018 in IJEAST (http://www.ijeast.com)

Weiss, 2012)(Chen, et al., 2017)(Amos, Turner, & White, 2013). They are broadly classified into two categories: Signature based detection system and anomaly based detection system. The signature based dataset intrusion detection system discovers malicious activity in the mobile on the basis of some predefined signatures. This system constructs a unique signature for the malware and detects malware by matching the signature with the collected data. The major limitation to this approach is that it cannot handle such type of attacks which are not in its predefined list. With the rapid change in the mobile technology, easy availability of mobile application, it is required to update the malware signature database frequently, which leads to a tedious task for the database managers. On the other hand, anomaly based intrusion detection system uses machine learning classifiers(Amos, Turner, & White, 2013). It does not work on the basis of previous knowledge rather it detects malware by learning from their behavior. This type of malware detection system, continuously monitors the different features obtained from the system and then applies machine learning classifiers to classify the collected and observed data into benign or malicious activity. It works in two phase: Training phase and testing phase(Shabtai, Kanonov, Elovici, Glezer, & Weiss, 2012). In training phase the machine is provided with the dataset consists of benign (normal) and malicious (abnormal) data. Machine monitors and capture the behavior of the system in case of normal operation as well as in abnormal operation. And on the basis of these feature set, learning algorithm generates a trained classifier.
In Testing Phase, a different set of collected data consists of both benign and malicious data, is provided to machine. The trained classifier will classify the data into normal or abnormal class.

III. IMPORTANCE OF FEATURE SELECTION IN MACHINE LEARNING

The feature selection process is an important process in high dimensional data mining applications(Mukherjee & Sharma, 2012). The subset of features is selected before applying to the learning algorithm. Here are some of the benefits of using feature selection method:

- Data reduction makes the better visualization of the trend in the data.
- This approach handles the noisy and irrelevant data from the dataset, so that one can get more accurate results.
- It requires a lot of efforts (in terms of training time, experiment cost) to analyze a large amount of data, with lots of features, while by applying feature selection methods, we can get a subset of important features that will actually effect the result of the experiments.

Adequate selection of features may improve the efficiency of the classifier. The Feature selection methods are broadly classified into the following: Filter method and Wrapper method(analyticsvidhya.com).

In Filter Method, process of selecting the features is independent of the learning algorithm. Rather, it uses the simple correlation methods for the prediction of the attributes. Chi square test, information gain etc. are the techniques used under this category.

In Wrapper Method, a random subset of features is taken and used to train the model. Based on the inferences drawn from the previous model, it is decided to keep the feature or to discard the features from the subset. Some of the common techniques used for this purpose are: Forward Selection: Initially, we'll start with the zero features in the subset and with each iteration, we keep on adding the features to our subset till the increase in performance. And we'll stop this process at the stage where we don't find any improvement in the performance. In Backward Elimination, we start with all the features in the subset, and remove the least important feature with each round. We repeat this process of removing features until we don't observed any improvement in the performance.
Recursive Feature Elimination, It recursively creates the models with the limited features. A new subset of features is chosen and model is trained. And with each iteration, the performance of each model is compared. This technique aims at finding the best performing feature subset.

IV. PROPOSED METHODOLOGY

This section presents the overall workflow of the proposed model. There are two phases: first, data collection and second, feature selection and extraction.

International Journal of Engineering Applied Sciences and Technology, 2018
Vol. 3, Issue 3, ISSN No. 2455-2143, Pages 56-61
Published Online July 2018 in IJEAST (http://www.ijeast.com)

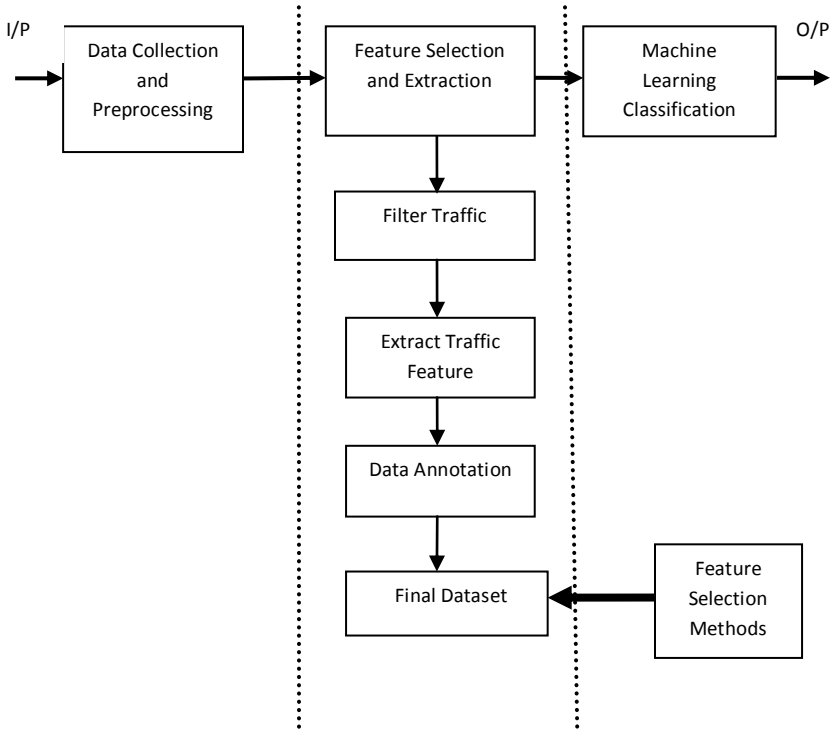

Figure 1. Flowchart of the Proposed Methodology

In the first phase, data is collected from various benign as well as from malicious applications. We have used a sample of dataset used in(Narudin, Feizollah, Anuar, & Gani, 2016).

The second phase is the feature selection and extraction phase. Various tools are available to monitor the nature of the traffic. Tools like wireshark (http://www.wireshark.org)are open source and freely available. It is used to analyze the network traffic and one can easily filtered out the desired packet type. And after the filtration process, it is easy to extract the features of the packet. Now, the extracted features are stored in the database which needs to be used in the next phase. Then this stored information is used to train the machine learning classifiers for the malware detection.

Besides all, it is challenging task to find out the most relevant features which give the optimum information about the application being malignant or benign. Working with large number of features leads to redundancy. This further leads to increase in processing time and reduces the accuracy of the model.

V. EVALUATION MEASURES

To evaluate the performance of the detection system, it is necessary to choose performance metrics. Results of an experiment can be stored in the form of table, which is known as confusion matrix(Davis & Goadrich, 2006). The following measures were derived from the confusion matrix:

- *True Positive (TP)*: It is the number of correctly classified instances from the dataset as positive. As TP increases, we get the better results.
- *False Positive (FP)*: It is number of normal samples from the dataset which are classified as malware. The decrease in FP, results in the more accuracy of the system.
- *True Negative (TN)*: It is the number of malware samples classified correctly.
- *False Negative (FN)*: It is number of malware samples classified as normal.
- *True positive rate (TPR)*: It is the value of predicted malware classified correctly. It is computed in equation 1:

$$TPR=TP/(TP+FN) \quad (1)$$

- *False Positive Rate (FPR):* It is the value of predicted normal data to malware or incorrect prediction. It is calculated as in equation 2.

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

- *Precision:* It is the positive predicted rate. And calculated as :

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

- *Recall:* It is equivalent to True positive rate. It gives the percentage of the samples classified correctly.

- *F-Measure:* With this measure, the performance of the complete system can be analyzed by combining precision and recall into one single formula:

$$F\_Measure = 2 * Recall * Precision / (Recall + Precision) \quad (4)$$

- *ROC Curve:* The receiver operating characteristics curve is known as ROC curve. The area below the ROC curve is known as Area under curve(AUC), is used for the analyzing the performance of the method.

International Journal of Engineering Applied Sciences and Technology, 2018
Vol. 3, Issue 3, ISSN No. 2455-2143, Pages 56-61
Published Online July 2018 in IJEAST (http://www.ijeast.com)

VI. EXPERIMENTS, RESULTS AND ANALYSIS

This section presents the experimental results and the performance evaluation of different feature selection techniques. In our experiment, we have taken the sample dataset(Narudin, Feizollah, Anuar, & Gani, 2016), and apply naïve bayes classification. The reason for choosing naïve bayes is, it is simple Bayesian probability model (Lewis, 1998). It works on independent assumptions. It assumes that the probability of one attribute does not affect the other. Weka 3.6(Hall, Frank, Holmes, Reutemann, Pfahringer, & Witten, 2009) on windows operating system is used to perform the experiments.

We have analyzed the performance of the classifier using three different feature selection methods of Weka. The methods used in the experiment are cfsSubsetEval and classifierSubsetEval with Random Search technique and PCA(Principal component analysis)(Virmani, Taneja, Chawla, Sharma, & Kumar, 2016) with ranker search technique. For the evaluation, we have used 10-fold cross validation and full training set. The experiment is carried out on a sample of dataset which consists of 11 features. Table 1 shows the performance of the classifier without feature selection method.

Table 2 shows the number of attributes is selected on applying different feature selection methods. Table 3 presents the average weighted measures on applying different feature selection techniques.

Table1: Performance evaluation without feature selection

Classifier Used	Feature Selection Methods	TPR (%)	FPR(%)	Precision	Recall	F-Measure	ROC Area
Naïve Bayes	CfsSubsetEval	0.6	0.4	0.6	0.6	0.6	0.57
	Classifier SubsetEval	0.6	0.4	0.61	0.6	0.58	0.8
	PCA	0.7	0.3	0.70	0.7	0.69	0.62

Table2: Summary of number of features selected by different feature reduction method

Naïve Bayes	TPR (%)	FPR(%)	Precision	Recall	F-Measure	ROC Area
	0.6	0.4	0.6	0.6	0.6	0.6
	0.6	0.4	0.6	0.6	0.6	0.54
Average Weighted Measure	0.6	0.4	0.6	0.6	0.6	0.57

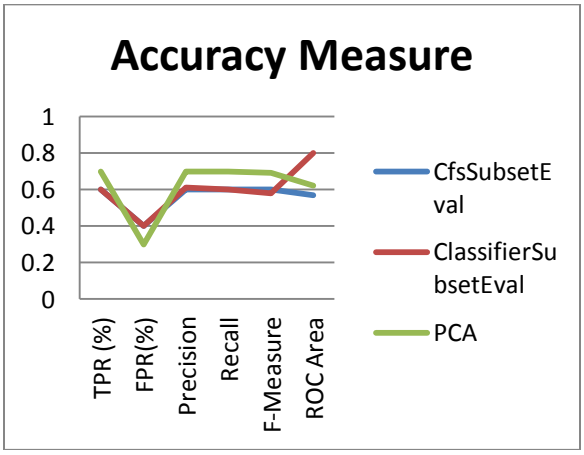

Figure 2: Performance Comparison of SubsetEval, ClassifierSubsetVal and Principal Component Analysis with Naïve Bayes

Table 3: Performance evaluation with feature selection

Feature Selection Technique used	No. of attributes selected	Selected Attributes
cfsSusetEval + Random Search	5	2,3,4,6,11
PCA +Ranker	4	1,2,3,4
ClassifierSubsetEval + random Search	6	1,3,6,7,8,11

Figure 2 shows the comparative graph for the classification accuracy with various feature selection methods. Experimental results show that feature subset identified by the ClassifierSubsetEval outperforms. From Table 3, it is observed that Naïve Bayes with Cfs SubsetEval, ClassifierSubsetVal and PCA have given promising results. TPR for Cfs SubsetEval, ClassifierSubsetVal and PCA is 0.6, 0.6 and 0.7; F-measure is 0.6, 0.58 and 0.69; ROC Area is 0.57, 0.8 and 0.62. The results clearly indicate that the ROC Area has shown positive results with Classifier SubsetEval ;and PCA. PCA with Naïve

Bayes gives best recall with 0.7. The performance evaluation interms of accuracy for the feature selection methods SubsetEval, ClassifierSubsetVal and Principal Component Analysis is given in Figure 2.

VII. CONCLUSION

In this paper, various feature selection methods are analyzed for filtering the features in malware detection system. As discussed in section IV, the mobile applications are governed by features which give useful insights about the application being malignant and benign. The proposed methodology uses effective feature selection method SubsetEval, ClassifierSubsetVal and Principal Component Analysis with Naïve Bayes classifier. The results given in Table 1 and 3 suggest that the method is useful and has given consistent results. As it can be observed that the performance of Naïve Bayes without feature selection gives a weighted average of 0.57 for ROC curve. The same classifier with ClassifierSubsetVal and Principal component analysis gives ROC curve as 0.8 and 0.62 respectively. The result so obtained clearly indicates that the proposed method *with feature selection* outperforms the method without feature selection.

VIII. REFERENCES

(n.d.). Retrieved from http://www.wireshark.org.

(n.d.). Retrieved from analyticsvidhya.com: https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/

Amos, B., Turner, H., & White, J. (2013). Applying Machine learning classifiers to dynamic android malware detection at scale. *IWCMC 2013* (pp. 1666-1671). Sardinia, Italy: IEEE.

Chen, Z., Yan, Q., Han, H., Wang, S., Peng, L., Wang, L., et al. (2017). Machine learning based mobile malware detection using highly imbalanced network traffic. *Elsevier , 433-434*, 346-364.

Coronado-De-Alba, L. D., Rodríguez-Mota, A., & Escamilla- Ambrosio, P. J. (2016). Feature Selection and Ensemble of Classifiers forAndroid Malware Detection. *Communications(LATINCOM).* Medellin, Colombia.

Davis, J., & Goadrich, M. (2006). The Relationship Between Precision-Recall and ROC Curves. *International Conference on Machine Learning.* Pittsburgh.

Feizollah, A., Anur, N. B., Sallah, R., & Wahid, A. (2015). A review on feature selection in mobile malware detection. *Elsevier* .

Hall, M., Frank, E., Holmes, G., Reutemann, P., Pfahringer, B., & Witten, I. H. (2009). The WEKA data mining software: an update. *ACM SIGKDD.* ACM.

Ham, H. S., & Choi, M. J. (2013). Analysis of android malware detection performance using machine learning classifier. *ICTC 2013.* Jeju, South Korea: IEEE.

Kohavi, R. (1996). Scaling up the accuracy of Naive-Bayes classifiers: a decision-tree hybrid. *KDD.* Citeseer.

Lewis, D. D. (1998). Naive Bayes at Forty: The Indepnedence assumption in Information Retrieval. *ECML* (pp. 4-15). Springer.

Milosevic, N., Dehghantanha, A., & Raymond Choo, K.-K. (2017). Machine learning aided Android malware classification. *Elsevier , 61*, 266-274.

Mukherjee, s., & Sharma, N. (2012). Intrusion Detection using Naive Bayes Classifier with Feature. *Elsevier* , 119-128.

Mukherjee, S., & Sharma, N. (2012). Intrusion Detection using Naive Bayes Classifier with Feature. *Elsevier* , 119-128.

Narudin, F. A., Feizollah, A., Anuar, N. B., & Gani, A. (2016). Evaluation of machine learning classifiers for mobile malware detection. *20* (1), 343-357.

Shabtai, a., Kanonov, u., Elovici, Y., Glezer, C., & Weiss, Y. (2012). "Andromaly": a behavioral malware detection framework for android devices. *Elsevier* , 161-190.

shabtai, A., Moskovitch, R., Elovici, Y., & Chanan, G. (2009). Detection of malicious code by applying machine learning classifiers on static features: a state of art survey. *Elsevier , 14* (1), 16-29.

Shabtai, A., Tenenboim-Chekina, L., Mimran, D., Rokach, L., & Shapira, B. (2014). Mobile malware

detection through analysis of deviations in application network behavior. *Elsevier* , 1-18.

Sung, A. H., & Mukkamala, S. (2004). The Feature selection and Intrusion detection problems. *Annual Asian Computing Science Conference* (pp. 468-482). Springer.

Virmani, D., Taneja, S., Chawla, T., Sharma, R., & Kumar, R. (2016). ENTROPY DEVIATION METHOD FOR ANALYZING NETWORK INTRUSION. *Computing, Communication and Automation (ICCCA2016)* (pp. 515-519). IEEE.