Mobile Malware Detection using Anomaly Based Machine Learning Classifier Techniques

A.Hemalatha, Selvabrunda

Abstract— Mobile phones are a significant component of people's life and are progressively engaged in these technologies. Increasing customer numbers encourages the hackers to make malware. In addition, the security of sensitive data is regarded lightly on mobile devices. Based on current approaches, recent malware changes fast and thus become more difficult to detect. In this paper an alternative solution to detect malware using anomaly-based classifier is proposed. Among the variety of machine learning classifiers to classify the latest Android malwares, a novel mixed kernel function incorporated with improved support vector machine is proposed. In processing the categories selected are general information, data content, time and connection information among various network functions. The experimentation is performed on MalGenome dataset. Upon implementation of proposed mixed kernel SVM method, the obtained results of performance achieved 96.89% of accuracy, which is more effective compared with existing models.

Keywords: Machine Learning, Malware detection, Mixed kernel function, Support vector machine

I. INTRODUCTION

The huge developments in the use of mobile and smart devices have led to a continuous increasing amount of Internet users. In a Google research, conducted by TNS Infra test GmbH shows that mobile devices have a monthly increase in ownership. For instance, in Japan, the figure rose from 6% in 2011 to 17% (Survey 2013). This technology provides various conveniences for end-users, by facilitating unlimited communication. Activities like social conferences, online videos and games are seen from anywhere and at any time through mobile devices. However, the increasing number of users also enables hackers to generate several malicious applications. In 2013, Symantec showed that households with malware had increased significantly by 58 percent compared to 2011, android was the largest target of the mobile platform in 2012 (García-Teodoroetal.2009). In September 2013, Symantec also published its report on transom ware growth across Android (Symantec 2013). Android is the most commonly attacked as it is an open-source operating system compared with all other operating systems on mobile devices (Teufl et al. 2014). Many applications, such as SlideME (2012), are acquired from non-official markets, although users are designed to download and set up Android applications.

In March 2011, it was found that 50 Android marketing applications had been subsequently eliminated due to the infection of DroidDream malware (Burguera et al. 2011). In April 2013, the researchers found the malwares operating in Google Play for at least 10 months, have been infecting 35 applications. Google has revealed that 2 to 9 million of infected applications had been performed (Arstechnica 2013). An F-secure report revealed that Android accounted 79% of the malwares in 2012, compared with 66.7% in 2011 and only 11.25% of the malware in 2010 (F-Secure 2013). Adware and data theft are the most common of malicious apps (Hardwarezone, 2013). Symantec discovered a malware package in June 2013 which ransomed user data. The latest ransomware takes $100 to unlock the device.This latest ransomware was produced by the same author in June 2013. Thus, it extends beyond just stealing user-data — the risk of portable malware is also an ransomware. In this article mobile bots are concentrated because 93 percent of the MalGenome specimens were networked bots (Yajin, Xuxian 2012).

In general, Mobile devices adopts traditional antivirus approaches against malwares. This is not so effective against mobile device malware (Sohretal.2011) as it contains continuously updating signature database and constantly modifying mobile malwares in order to bypass the multiple detection techniques.

For example, the first version of Droid Kung Fu came out in June 2011 and seemed to be one of the most advanced types of malware at the time. Shortly afterwards in July and August, the second and third variants emerged. In October the fourth version was found and shortly thereafter the fifth. Variants tend to use various encryption keys in order to safeguard themselves. The adaptation of malware indicates a continuous attempt by hackers to bypass the detection process (Yajin and Xuxian 2012). As a result, mobile devices may pose other threats including spam, private information and user credentials theft, contents management and unauthorized remote access, following the use of anti-virus defense mechanisms. The SMS is also in imminent danger as text messages are sent without security concerns.

An intrusion detection system (IDS) provides an early detection and the prospect of warning users to reduce the threat. Traditional approaches like antivirus can not be detected soon and most malware such as bots populates mobile phones well before they are detected (Garcia-Teodoroetal. 2009). An antivirus also needs to carefully track the operations of a device and regular signature updates to identify malware. This requires an excessive

Revised Manuscript Received on September 10, 2019.

A.Hemalatha, Research and Development Centre, Bharathiar University, Coimbatore, Tamil Nadu, India.
(E-mail: hema.phd.research@gmail.com)
Selvabrunda, Computer Science Department, Cheran College of Engineering and Technology, Coimbatore, Tamil Nadu, India.
(E-mail: brindhaselva@yahoo.com)

amount of memory and energy, degrading the efficiency of mobile devices (Lai and Liu 2011). The purpose of the article therefore is to assess the efficacy of malware detection IDSs based on anomaly via network traffic. There are two kinds of IDS detection methods: misuse and anomaly (Shamshirband et al. 2013). In contrast to the misuse, intrusion detection does not require the anomaly-based IDS signatures. In addition, unidentified attacks based on comparable intrusion behaviour, can be identified by an anomaly-based IDS (Curiac and Volosencu 2012).

Classifiers of machine learning (ML) have been involved in the development of intelligent systems for several years. MLs acquire a labeled dataset and generate an output model which is able to process new data. Classifiers learn the input and labeled output from an abundance to construct a model. The adoption of machine learning classificators has therefore been demonstrated to improve the precision of detection (Anuar et al., 2008). The aim of this work is subsequently to assess classifiers for master learning so as to offer an alternative solution to malware detection using network traffic. In this study, we are able to capture network traffic from outside mobile devices to add advantages over built-in applications such as antivirus solutions that drain device resources. The detection method can also be performed via cloud services like Patel et al. (2013), which will analyze network traffic remotely. This strategy reduces software complexity and resource usage (Oberheide et al. 2008). It is common knowledge that the choice of the kernel feature of the vector support device has a major impact on the SVM classification outcome[20]. In this respect, the single kernel function used in the past is no longer suited for the classification of malware. This article suggested a new blended kernel function to classify mobile malware for an enhanced vector support system. In order to evaluate IDS efficiency in mobile application detection three different classifiers in this research: Byes Network, MLP, random Forest and Multi-kernel Proposed SVM are assessed.

This article provides a full research on the efficiency of the adoption of classifiers for the detection of malware in mobile apps. As the Android Operating System is used by end-users as the most common system, this study is focused on Android apps. The following are the contributions made to this work:

a) The assessment study included 1000 samples from the Android Malware Genome Project (Yajin and Xuxian 2012), which is regarded as an extensive sample on mobile devices of malware data.

b) Three classifiers were assessed from various parts of the assessment.

c) In the evaluation study, 11 features of this experiment were removed from the mobile applications ' network traffic.

d) The efficacy of the suggested classification is shown in an empirical validation undertaken.

The next chapter deals with job on the subject in this article. Section 3 describes the methodology and the malware detection overview carefully. Section 3 includes three primary stages: information collection, extraction and classification of machine learning. The experimental findings are presented in Section 4. Finally, section 5 includes the conclusion observations and suggestions for future studies.

II. LITERATURE REVIEW

Signature and anomaly detection detect general IDS intruders (Verwoerd & Hunt 2002). Signature-based methods detect intrusions by means of a predefined roster of recognized assaults. Although this option allows malware detection on a mobile app, the predefined identity database needs to be updated. Moreover, the rapidly changing element of mobile malware (Droide Kung Fu), by means of a method based on signature, is less efficient in detecting illegal activities. Because of Yajin and Xuxian (2012), up to 79.6 percent of mobile botts are available in our MalGenome dataset. However, anomaly-based machine learning approach is more than 90 times more sensitive. On the other side, an anomaly-based IDS technique uses classification devices and enables malware to be detected by studying their behavior. Malware comportement can be modelled using a classifier of computer training and using the manufactured model to identify fresh malware. For several years, classifiers of machine teaching (ML) have been involved in the development of smart structures through teaching computers for decision-making. ML constructs a model for new information, which identifies patterns of similarity with a information set labeled as an entry. A similar approach has been adopted by several studies with significant results for the purpose of effectively detecting intrudy (Sangkatsanee et al. 2011 ; Zhao et al. 2012).

In order to optimize the method of malware classification, specific malware properties should be collected–something that can only be accomplished through the investigation of malware operations. Two methods of malware analysis are available: the static and the dynamic (Egele et al. 2008). Static analysis is a malware examination process without mobile applications being carried out. For example, a survey was performed to inspect requested permissions to identify certain Android malware (Huang et al. 2013). Similarly, mobile threats against domestic security were identified via static analysis (Seoetal. 2013).

Static analysis is easy to implement, but it produces less information, which limits the extraction of potential malware features. Moreover, attackers use various methods, such as obscure code, for static analysis to avoid detection. As a consequence, dynamic analysis can be carried out and track the behavior of a mobile app to obtain helpful data and enhance functionality. There are several kinds of behaviors that can be found using vibrant assessment, such as the implementation (Burguera et al. 2011), system calls (Dinietal.2012) and produced traffic and used storage.

Sarma et al. (2012) published a research work in 2012 in which they analyzed permissions of Android files. They looked at over 150,000 applications and determined that 68.50 percent of normal applications require network access, with 93.38 percent requiring network access. In the implementation permissions for detection of anomalies, Huang et al. (2013) employed a static analysis and assessed four classifiers (i.e. Adaboost, NB, DT48 and SVM). The Malware rates for the Naïve Bayes classification were identified at 81 per cent. Therefore, they argued that permission-based analysis is a fast filter to identify abnormal applications. The permit-based analysis, however, is less effective for malware such as Base Bridge, which hides an updated version in the original application so that it slips into a mobile device without the knowledge of the user and circumvents permission. Therefore, we have chosen to use anomalous techniques and dynamic analysis in this document because of weaknesses in signature-based technology and static analyses.

The dynamic analysis scheme Crowdroid, which examines implementation conduct to identify malware in Android via machine calls, was suggested by Burguera et al. (2011). This frame can detect the malware that is self-written (i.e. PJApps and Hong Toutou). Similarly, Shabtai et al. (2012) used the Andromaly structure to identify the finest technique of classification from six classifier groups, DTJ48, NB, K-Means, histogram and logistic regression. The frame uses functional selection methods, such as chi-square, fishing score and information gain to improve detection precision. As a result, the decision tree (i.e. J48) classification and the method of gaining information achieved a 99.7 percent accuracy rate. Although Shabtai et al. (2012) were sadly very precise, they have tested their structure using auto-written malware to achieve unrealistic outcomes. In this work instead of self-written malware, we used 1.000 actual malware.

Dinietal.(2012) suggested a multilevel error detection design by mixing two machine call concentrations: core and user level. Dini et al. effectively achieved a 93% precision level for the 10 malwares with 12 system call sate features together with the classifying K-nearest neighbours (KNN). This strategy is promising but cannot detect malware that prevents root-approved system calling, e.g. SMS malware that is invisible within the kernel (Security 2011). In this work, network traffic analyses are used to compensate for the weakness of system calls.

Zhao et al. (2012) have proposed to detect unknown malware in mobile devices based on the SVM Machine learning classification. The concentrate was on leakage of data protection data and concealed deposit. They assessed three kinds of malware: Gemini, Droid Dream and Plankton. This framework is therefore limited to few types of malware and more would be necessary to enhance detection accuracy.

A dual smartphone protection framework was presented by Su et al. (2012), which consists of two phases: verification services and a network monitoring tool. The first, the check service stage, is used by the system call statistics to distinguish between malicious and normal codes in an application. Phase two monitors possible malicious codes identified in phase one. The J48 and random forest are the two simulated classifiers to evaluate the proposed framework. The classification of the random forest was 96.67% and the J48 classification reached 91% precision in the identification. However, 32 malware households with 180 specimens compared to 1,000 specimens used in this study were used in this method. Additional examples contribute to better results, since it offers more extensive and reliable results, including as many samples as necessary.

In 2013, STREAM (Amos et al. 2013) was implemented to collect a amount of batteries, storage, network and authorisation functions. It applied a number of machine learning classifications on the data collected. The authors used Android emulators to collect selected characteristics, which have proven to be not so accurate as a real device (Raffetseder. 2007), and claimed it was not possible to use a real device. We used a real device in this work to perform part of our experiment.

In 2013 Hyo-SikandMi-Jung(2013) published his study in which machine learning was applied to selected features of Android apps. The vulnerability of their work is that they have assessed their suggested scheme for five different malware households, while the 49 malware groups have assessed our scheme.

DREBIN (Arp et al. 2014) was published in 2014, conducting an extensive analysis of Android malware detection applications. This collects allowance, hardware access, API calls, network addresses, etc. However, malware using obscuration technology and dynamically loaded code technology cannot be detected, while our job is completely capable of identifying such malware.

Thus, these approaches show the reason behind the application of machine learning classifications to detect mobile malware. However, the previous assessment is restricted to a certain quantity of malware, and function choice in the classification method is considered by few solutions to increasing results precision.

The precision rate plays an important role in measuring the effectiveness of the approach in intrusion detection systems. One motivation for this study is to increase the truly positive rates and to lower them beyond other studies. The experiments with machine learning classification successfully achieved better detection rates, based on the results section.

III. PROPOSED METHODOLOGY

This section presents the workflow of the two experiments. The experimental workflow design of three stages is illustrated in Figure1.

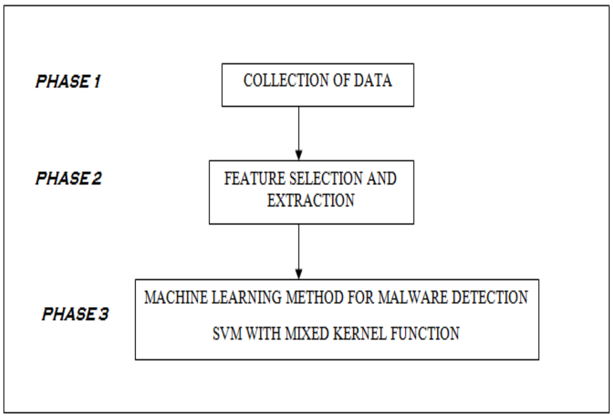
PHASE 1
COLLECTION OF DATA
PHASE 2
FEATURE SELECTION AND EXTRACTION
PHASE 3
MACHINE LEARNING METHOD FOR MALWARE DETECTION
SVM WITH MIXED KERNEL FUNCTION

Figure 1. Proposed Methodology

The first stage is the collection of data that collects the network traffic and passes it on to the next phase. TCP packets shall be filtered during the second phase, the feature selection and extraction, and functions chosen from the various network features are extracted, labelled and stored in a next phase database. The machine learning classification entails the last phase, in which the information in a database forms the classification machine to produce a detection model. Each phase is described in the following sections.

3.1 Data Collection

The selected classifiers were evaluated by public and private data bases. MalGenome is the public dataset of 1260 malware packages collected from2010to2011.It contains some of the most malicious mobile malware in the world. Surprisingly, there was a collection of 14 out of 49 families from the Google Play official Android market. A collection of the most recent mobile malwares is the private dataset. As mobile malware changes continue, which is the hacker's strategy to circumvent the current detection methods, as well as the rapids in the number of mobile malwares, our security team collected 30 malware samples and was monitored by a malware analyst.

MalGenome Project (2013) includes 1,260 malware information samples in 49 households, 1,000 of which were selection for their network model from 49 malware households. It is because 93 per cent of all the samples are bots that 1000 samples are selected. Online dynamic analysis platforms, namely Anubis ISEC lab Anubis (2013) and Sand Droid (2013), were used for the samples analysis. In a regulated setting, dynamic internet analysis systems operate an algorithm and record networking traffic, which is used in studies. In the MalGenome sample set of 1,260 samples, 1,007 traffic samples were produced by Anubis isec Lab, and 164.Thesedynamic analytic platforms capture network traffic in a controlled setting. Given that malware samples do not require traffic installation on a physical device, this approach reduces the time required to generate traffic. 93 percent of the 1,260 MalGenome files are bots, which represent 1171 files. During our test, we verified for legitimate servers and deleted 171 files from unsuccessful servers.

3.2 Feature Selection and Extraction

During the data collection phase, wireshark and Java filtered benign and malware network traffic to remove unwanted packages from the collected packets. The routine Domain name scheme streams (DNS) were filtered out of the network traffic data; only TCP streams have then been used as a dataset since TCP streams transmit conversations between mobile malware and hackers. After that, all associated information was extracted using tshark (2013), a terminal version of wireshark that used to extract 11 functionality from the TCP packets (table 2). The characteristics of the intrusion data set TUIDS (Gogoi2013), have been selected from a wide array of network characteristics. The primary task with regard to the choice of characteristics was above all to find the most important characteristics leading to the greatest true positive rate. A wide range of data set features should be filtered and refined. Furthermore, certain characteristics are correlated and hamper the method of intrusion detection. In addition, certain functions may include redundant data from other characteristics. Redundant characteristics improve time and decrease IDS exactness.

A technique named Weka computer training instrument called ClassifierSubsetEval was used to select the variable. After the application of the selection algorithm depending on the outcomes, we chosen six out of 11 functions. It should be noted that no characteristic can differ between harmless applications and evil applications, but a set of characteristics. A combination of all the features selected helped find anomalies, rather than just one feature.

In China, particular IP addresses are linked to malicious activities, for example. Moreover, the length of the frame is larger than normal frames due to leaked data. This leads to an anomaly detection with the mixture of both characteristics. Classifiers examine the applications from a group of functions and create patterns of behavior to detect malware. Every model of conduct, which deviates from a smooth and ordinary model is considered malignant.

The features chosen for our data set are specifically derived from four categories of features (Lee and Stolfo 2000). The classifications are fundamental characteristics, depending on material, moment and relation. The most common network traffic function used by several malware tracking scientists is the fundamental function category. This section provides basic network traffic information such as the IP address of source, IP address of destination, host number, frame number of destination port, and frame numbers. The second category of content-based features is a common link record pattern. The time-base function in the third category records a connection with the same host as the current connection in the past 10s. The 4th cluster or link-based characteristics count the amount of data packets continuously from origin to target and vice versa. In this research, 11 chosen characteristics have been included in our malware tracking data set as stated above.

The obtained characteristics were saved as a series of CSV files. Each record consists of 11 network features summarized data. The intrusion-detection experts have labelled the data set as' normal' or' infected' with a malware-based dataset categorized as' infected' and a benign dataset as' normal,' finally using the malware-based data set with the normal dataset. By using the function selection method, we can select k from the extracted features for android application package files: Information Gain in equ 1.

$Gain\ (S,A) = Entrophy\ (S) - \sum \left|SV\right|\ Entrophy(SV)$
$V€Values(A) \left|S\right|$ (1)

This method depends on the entropy of the characteristics and chooses the highest profit value as the best function. A feature A is gained from a set of S instances.

3.3 Machine Learning Approach for Malware Detection

The characteristics are colleited in the signature database and split into training data and test information and are used for the detection of Android malware apps through conventional machine learning methods. In the final phase, the phase of the classifying of machines, the result of the classifiers is generated. The finest learning classification machine for malware detection is determined by these phases. This section presents the concepts and descriptions used in the present experiment of our selected classifier.

3.3.1 Random forest

The random forest is a mixture of Tin Kam's (1998's) random subspace and bagging method. Classifiers using many decision tree designs, random forest sets. To track each tree a distinct subset of training data is chosen. The rest of the training data are used to assess mistake and variability (Breiman 2001). This classification is a logic-based method which proves to generate a high-precision malware tracking outcome, such as Eskandari and Hashemi (2012); Su et al. (2012).

3.3.2 MLP

Multi-layer perceptron is an artificial neural network template for moving forward. MLP comprises of several node levels interacting via weighted links (Pal and Mitra 1992).

3.3.3 Support Vector Machine

The Vladimir Vabnik and Alexey Shervonenikis support vector machine (SVM) algorithm was invented in 1963. In 1993 Corrina Cortes and Vabnik developed and published an approved algorithm, which is currently in use.

In order to carry out various data operations including ranking procedures SVM algorithm is the basis of data analysis in the learning algorithm. Data input into the classification process takes the form of two different data classes for the formation and learning phases of the algorithm, so that each type of data can be classified into its own class [7].

The algorithm generates a linear framework to distinguish the characteristics of each category when modelling the information from characteristics axis, to allow a maximum gap between these characteristics. After the training process, the algorithm begins self-learning and classifies another part of the data according to the way the course is done. In addition to the ability of the algorithm to create a linear framework, nonlinear frames are created by using the kernel trick. SVM is one of the most popular automated classification methods among classification algorithms used in the machine learning field, depends on a curve or high level of separation between the data entered and its narrowness when classifying issues with binary categories. 1 in positive or-1 in negative samples.

This algorithm calculates a layer or object array in a different dimension, whose size differs from the vector characteristics size. The accuracy of the algorithm depends on whether it can separate both classes so that the sample closest to each side of the two kidneys, known as the margin, is distant. Generally, the higher the separation range, the lower the mistake of generalizing part of the data that is not specific to the phase of training[8]. Although the issue appears to be easy, classes can not often be separated in a linear way and then the axes of the properties vectors are moved to higher dimensions to be divided by a surface. The standard multiplication of vectors is calculated from this point of view by the matrix function, in which a number of points are determined by the separator surface. The product of its standard multiplication by a vector (in the highest dimension) in the new coordinates is constant.

One of the most popular computer training procedures is the ranking of information. The objective is to categorize a new point with data points of a two-class type and to determine which class belongs to. It is considered that the item of the statement has the number of attributes of the AN, and if the separation area dimension is smaller than AN by 1 the classification is linear and otherwise not linear If there are more than one separating line, this will ensure that the distance between the two nearest points of two different classes, the so-called hyper-plane, is wider.

SVMs cannot find a linear hyperplane that can insulate the data in two classes for many events. This issue can be addressed using a specific non-linear function, to convert the data located in a high dimensional space. The data can be insulated depending on this procedure in such a way that the space on which the data is placed can produce linear separable hyperplane. However, the calculation of inward results of two transformed victors in practice would be impossible, due to the high dimensionality of the space of the data. Using SVM kernel features This issue can be dealt with by using the transformed data points in feature space instead of the internal result [10]. The functional and accurate use of SVM kernel functions can essentially decrease the process by using a computer.

The highest choice of SVM kernels is essential for the efficiency of SVM evaluation. A compatible SVM kernel allows you to learn the SVM algorithm in the workout. The maximum value can be calculated in Eq for the optimal classification plane. (2) subject to the condition of constraint Eq. (3).

$$W(\alpha) = \sum_{ki=1=0}^{l} \alpha_i \sum_{ij} \alpha_i \alpha_j y_i y_j K(x_i x_j) \qquad (2)$$

Under the condition of constraint

$$\sum_{i=1}^{l} \alpha_i^0 y_i = 0, C \geq \alpha_i^0 \geq 0, i = 1, \ldots..l \qquad (3)$$

Where $K(X_i,x_j)$ is kernel function

The proposed paper mixed kernel function was defined in (4) as follows:

$$\mathrm{k}(x, x_i) = \lambda_1 * \exp\left(\frac{\|x-y\|^2}{2\sigma^2}\right) + \lambda_2 * \left((x.x_i) + 1\right)^d \quad (4)$$

The optimum classification function is achieved as (5) after the resolution of the optimum solutions corresponding to the above-mentioned coefficients.

$$f(x) = sgn\left(\sum_{i=1}^{l} \alpha_i^0\, y_i K(x_i, x) + b_o\right) \qquad (5)$$

IV. EXPERIMENTAL RESULTS

This section presents the test results and the malware detection performance evaluation. The public dataset evaluation experiment MalGenome. Meanwhile, WEKA used the most recent malware experiment as a test set. A training set consists of a set of data used in several fields to discover predictive relationship potential. The test set plays a key role in investigating the classifiers effectiveness. The test dataset does not appear in the training dataset. Finally, the optimal classification was determined following the test outcomes for both scenarios.

Evaluation measures

To effectively assess the tracking efficiency, the suitable efficiency metrics have to be identified. In order to calculate and apply the steps for a classification assessment, the following were extracted from the misunderstanding matrices. When the TPR is the properly classified malware valuation, or the true positive rate. FPR is the wrongly expected price valuation of ordinary information as malware. TP or true positive is the correct classification of the number of normal data. TN, true negative, is properly classified as the amount of malware samples. FP is the number of normal samples that are classified as malware, false positive. The number of malware specimens classified as normal is FN, the false negative. Precision, also known as positive-predictive value, yields the frequency of outcomes, not insignificant outcomes. Remember the awareness for the most important outcome. F-Measure is the utility that measures the full output of the scheme by adding accuracy and recall to a final amount. The maximum value of 1,000 shows the best outcome.

Table 1. Evaluation result of Malware detection

Classifier	TPR (%)	FPR (%)	Precision	Recall	F-Measure
RF	91.88	8.12	0.919	0.918	0.918
MLP	92.90	7.10	0.929	0.929	0.929
SVM	94.83	5.17	0.94	0.948	0.948

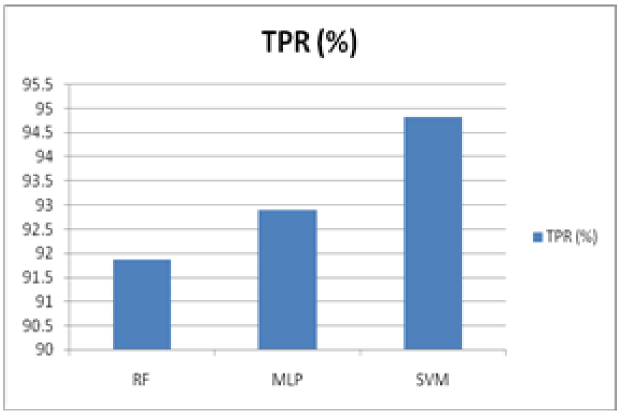

Figure 2. TPR of Classifiers

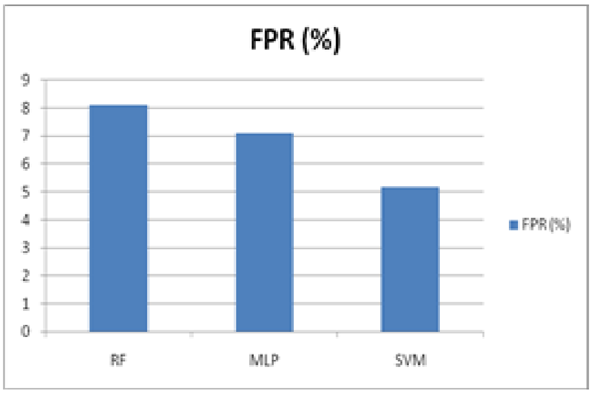

Figure 3. FPR of Classifiers

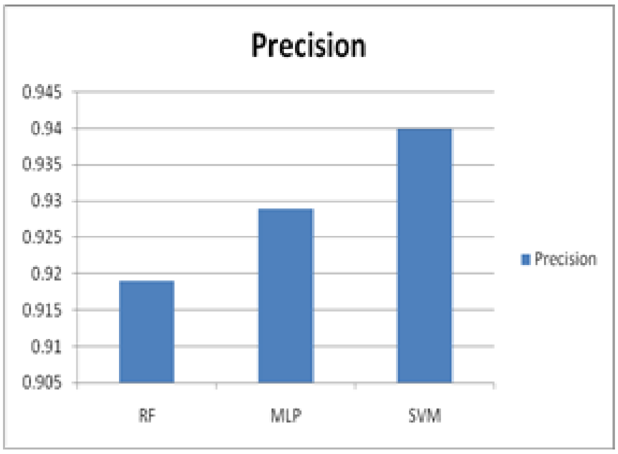

Figure 4. Precision of Classifiers

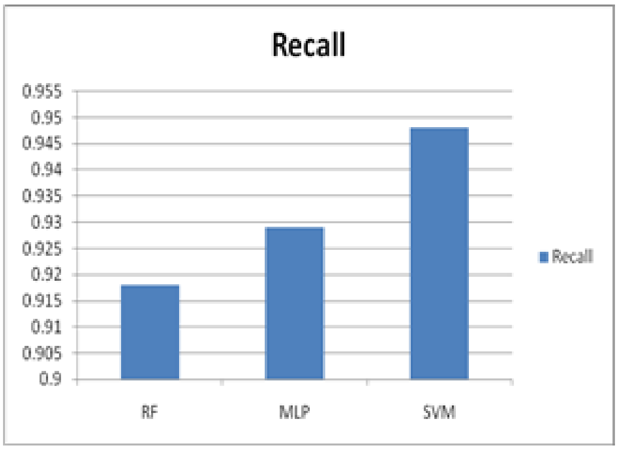

Figure 5. Recall of Classifiers

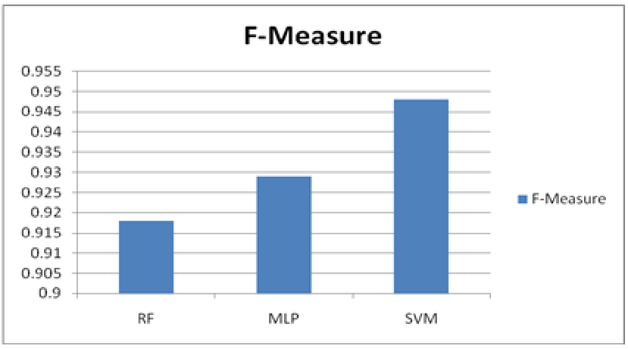

Figure 6. F-Measure of Classifiers

Table 1 and the figures 2,3,4, 5 and 6 presents the outcomes acquired by the tenfold validation test with five chosen classifiers. The table displays the performance of each classifier in the malware detection experiment. The performance of classifiers is measured by five measuring metrics: TPR, FPR, precision, reminder and f-measure. The random forest accomplished an efficiency of 91.8 percent, while the MLP reached an estimate of 92.9 percent, while the lowest tracking frequency in this research reached 94.8 percent by the proposed SVM classification.

V. CONCLUSION

This study has presented an evaluation using machine-learning classifiers to effectively detect mobile malware by choosing the appropriate networking features for classifier inspections, as well as to find the ideal classifier based on TPR values. The findings and achievement of the classifiers were overwhelming. We evaluate several machine learning classification systems in this research, to improve the malware detection result of a wide range of samples and to obtain the best classification capable of detecting mobile malware. Random forest, multi-layer perceptron and proposed SVM are the classifiers selected. Experimental results indicate 96.89 percent accuracy of the detection rate with current classifiers for the Genome Malware dataset. It also proves that the latest malware can be identified by machine classifiers. The larger the dataset, the longer the processing time is needed to build the detection model and increase the accuracy, retraction, and f-measurement. In practice, we are proposing the development of real-time mobile malware identification through machine-learning classifiers in the cloud. Nevertheless, this strategy has demonstrated that machine learning is effective and efficient in true malware operations.

REFERENCES

1 Amos, B., Turner, H., & White, J. (2013, July). Applying machine learning classifiers to dynamic android malware detection at scale. In 2013 9th international wireless communications and mobile computing conference (IWCMC) (pp. 1666-1671). IEEE.

2 Android (2013) Android 4.2, Jelly Bean. http://www.android.com/about/jelly-bean/.

3 Anuar, N. B., Sallehudin, H., Gani, A., & Zakaria, O. (2008). Identifying false alarm for network intrusion detection system using hybrid data mining and decision tree. Malaysian journal of computer science, 21(2), 101-115.

4 Anubis (2013) Anubis: analyzing unknown binaries. http://anubis.iseclab.org/.

5 Arp, D., Spreitzenbarth, M., Hubner, M., Gascon, H., Rieck, K., & Siemens, C. E. R. T. (2014, February). Drebin: Effective and explainable detection of android malware in your pocket. In Ndss (Vol. 14, pp. 23-26).

6 Arstechnica (2013) More Bad News for android: new malicious apps found in google play. http://arstechnica.com/security/2013/ 04/more-bad news-for-android-new-malicious-apps-found-in-go ogle-play/.

7 Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern recognition, 30(7), 1145-1159.

8 Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

9 Burguera, I., Zurutuza, U., &Nadjm-Tehrani, S. (2011, October). Crowdroid: behavior-based malware detection system for android. In Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices (pp. 15-26). ACM

10 Curiac, D. I., &Volosencu, C. (2012). Ensemble based sensing anomaly detection in wireless sensor networks. Expert Systems with Applications, 39(10), 9087-9096.

11 Dini, G., Martinelli, F., Saracino, A., &Sgandurra, D. (2012, October). MADAM: a multi-level anomaly detector for android malware. In International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security (pp. 240-253). Springer, Berlin, Heidelberg.

12 Egele, M., Scholte, T., Kirda, E., &Kruegel, C. (2012). A survey on automated dynamic malware-analysis techniques and tools. ACM computing surveys (CSUR), 44(2), 6.

13 Eskandari, M., & Hashemi, S. (2012). A graph mining approach for detecting unknown malwares. Journal of Visual Languages & Computing, 23(3), 154-162.

14 Felt, A. P., Finifter, M., Chin, E., Hanna, S., & Wagner, D. (2011, October). A survey of mobile malware in the wild. In Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices (pp. 3-14). ACM.

15 F-Secure (2013) Android accounted for 79% of all mobile malware in 2012, 96% in Q4 alone. http://techcrunch.com/2013/03/07/f-secure-android-accounted-for-79-of-all-mobile-malware-in-201296-in-q4-alone/. Accessed 1st June 2013

16 Gogoi, P., Bhattacharyya, D. K., Borah, B., &Kalita, J. K. (2013). MLH-IDS: a multi-level hybrid intrusion detection method. The Computer Journal, 57(4), 602-623.

17 Huang, C. Y., Tsai, Y. T., & Hsu, C. H. (2013). Performance evaluation on permission-based detection for android malware. In Advances in Intelligent Systems and Applications-Volume 2 (pp. 111-120). Springer, Berlin, Heidelberg.

18 Ibrahim, L., Salah, M., Rahman, A. A. E., Zeidan, A., &Ragb, M. (2013). Crucial role of CD4+ CD 25+ FOXP3+ T regulatory cell, interferon-γ and interleukin-16 in malignant and tuberculous pleural effusions. Immunological investigations, 42(2), 122-136.

19 Liang, S., Keep, A. W., Might, M., Lyde, S., Gilray, T., Aldous, P., & Van Horn, D. (2013, November). Sound and precise malware analysis for android via pushdown reachability and entry-point saturation. In Proceedings of the Third ACM workshop on Security and privacy in smartphones & mobile devices (pp. 21-32). ACM.

20 Wyatt, T. (2010). Security alert: Geinimi, sophisticated new android trojan found in wild. Online] December,2010.

21 Patel, A., Taghavi, M., Bakhtiyari, K., &JúNior, J. C. (2013). An intrusion detection and prevention system in cloud computing: A systematic review. Journal of network and computer applications, 36(1), 25-41.

22 Play G (2013) Shop android apps. https://play.google.com/store?hl=en.

23 Project MG (2013) Android malware genome project. http://www.malgenomeproject.org/.

24 SandDroid (2013) SandDroid-an APK analysis sandbox. http://sanddroid.xjtu.edu.cn/.

25 Sangkatsanee, P., Wattanapongsakorn, N., &Charnsripinyo, C. (2011). Practical real-time intrusion detection using machine learning approaches. Computer Communications, 34(18), 2227-2235.

26 Sanz, B., Santos, I., Laorden, C., Ugarte-Pedrero, X., Nieves, J., Bringas, P. G., & Álvarez Marañón, G. (2013). MAMA: manifest analysis for malware detection in android. Cybernetics and Systems, 44(6-7), 469-488.

27 Sarma, B. P., Li, N., Gates, C., Potharaju, R., Nita-Rotaru, C., & Molloy, I. (2012, June). Android permissions: a perspective combining risks and benefits. In Proceedings of the 17th ACM symposium on Access Control Models and Technologies (pp. 13-22). ACM.

28 Selvaraj, D., & Ganapathi, P. (2014). Packet payload monitoring for Internet worm content detection using deterministic finite automaton with delayed dictionary compression. Journal of Computer Networks and Communications, 2014.

29 Su, X., Chuah, M., & Tan, G. (2012, December). Smartphone dual defense protection framework: Detecting malicious applications in android markets. In 2012 8th International Conference on Mobile Ad-hoc and Sensor Networks (MSN) (pp. 153-160). IEEE.

30 Divya, S., & Padmavathi, G. (2014). A novel method for detection of internet worm malcodes using principal component analysis and multiclass support vector machine. International Journal of Security and Its Applications, 8(5), 391-402.

31 Survey G (2013) Our mobile planet: global smartphone user.

32 Symantec (2013) Android ransomware predictions hold true.

33 Midhunchakkaravarthy, D. D. (2016). An Efficient And Secure Detection Of Internet Worm Using Propagation Model. International Journal Of Innovations In Scientific And Engineering Research, 3(1), 8-15.

34 Teufl, P., Ferk, M., Fitzek, A., Hein, D., Kraxberger, S., &Orthacker, C. (2016). Malware detection by applying knowledge discovery processes to application metadata on the Android Market (Google Play). Security and communication networks, 9(5), 389-419.

35 tshark (2013) tshark-the wireshark network analyzer.

36 Verwoerd, T., & Hunt, R. (2002). Intrusion detection techniques and approaches. Computer communications, 25(15), 1356-1365.

37 Divya, S. (2013). A survey on various security threats and classification of malware attacks, vulnerabilities and detection techniques. International Journal of Computer Science & Applications (TIJCSA), 2(04).

38 Zhou, Y., & Jiang, X. (2012, May). Dissecting android malware: Characterization and evolution. In 2012 IEEE symposium on security and privacy (pp. 95-109). IEEE.

39 Yerima, S. Y., Sezer, S., McWilliams, G., &Muttik, I. (2013, March). A new android malware detection approach using bayesian classification. In 2013 IEEE 27th international conference on advanced information networking and applications (AINA) (pp. 121-128). IEEE.

40 Zhao, M., Zhang, T., Ge, F., & Yuan, Z. (2012). RobotDroid: a lightweight malware detection framework on smartphones. Journal of Networks, 7(4), 715.

41 Zheng, M., Sun, M., & Lui, J. C. (2013, July). Droid analytics: a signature based analytic system to collect, extract, analyze and associate android malware. In 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (pp. 163-171). IEEE.