Journal, Advances in Mathematical & Computational Sciences
Vol. 8 No. 3, September, 2020
www.mathematics-computationaljournal.info

Journal of Advances in Mathematical & Computational Sciences
An International Pan-African Multidisciplinary Journal of the SMART Research Group
International Centre for IT & Development (ICITD) USA
© Creative Research Publishers - Available online at
https://www.isteams.net/socialinformaticsjournal
DOI: dx.doi.org/10.22624/AIMS/MATHS/ V8N3P6
CrossREF Member Listing - https://www.crossref.org/06members/50go-live.html

SMSTDroid - Intrusion Detection System for Mobile Android Devices Payment System

[1]Ekong, U.O. and [2]Idio N.O.
[1,2]Department of Computer Science
Faculty of Science
University of Uyo,
Uyo, Akwa-Ibom State, Nigeria
E-mail: uyinomenekong@uniuyo.edu.ng, nancyidio24@gmail.com

ABSTRACT

Android has become the most popular Operating System for Smartphones and Tablets with an essential market share of about 70 to 80 percent. The open source structure of Android operating systems has introduced security vulnerabilities that can be readily exploited by third party applications. Cybercriminal take advantage of both the limited capabilities of mobile devices and the lack of standard security mechanism to naturally expand their malicious activities against Android mobile platform because of its huge market share. These hackers take advantage of the permission required on installation of application to develop mobile specific SMSTrojan malware specially design to subscribe to premium Short Message Services (SMS) numbers, intercept and delete the confirmation message thereby causing financial damage to the victims without their knowledge. To maximize profit, they continuously improve their creations to make them more and more resilient against anti-malware solution. Various techniques have been proposed for the detection of malware threats in Android and one promising set of techniques uses features extracted from permissions traces to infer malicious behaviours. In this paper, SMSTrojan malware behavior, the permissions required by these Trojans application on installation and its malicious intent on users' device financial charges are presented. More precisely, a framework for the detection of SMS-based Android malware on mobile payment system called SMSTDriod (Short Messaging Services Trojan for Android) is presented. The system employ the use of SVM to identify SMS-based malicious application. Performance of the malware classification and detection are evaluated against data collected from an open source malware repository using confusion matrix. The result yielded a detection rate of 0.84 true positive rate and low false alarm rate of 0.11 predicted by the model which implies that the number of misclassified malicious instances as benign was minimal.

Keywords: Malware, SVM, Trojan, SMS, Mobile Payment System.

Ekong, E.U. and Idio, N.O. (2020): SMSTDroid - Intrusion Detection System for Mobile Android Devices Payment System. Journal of Advances in Mathematical & Computational Sciences Vol. 8, No. 3. Pp 61-74 DOI: dx.doi.org/10.22624/AIMS/MATHS/ V8N3P6.
Available online at www.isteams.net/mathematics-computationaljournal.

1. INTRODUCTION

The emergence of mobile devices with increasing powerful computing capabilities and the permeating use of the mobile platform for developing sensitive applications (online banking, mobile commerce) has in no small measure improved computing and communication in recent times. Report by Gartner, (2016) shows a 78% increase in smartphone sales in 2016 with Android Operating System (OS) dominating the market. Nuremberg, (2019) and Mongardini J. and Radzikowski A. (2020) reports sale of 1.44billion units of smartphones in 2018 resulting in global sale of $522billion. Android is an open-source OS enabling publicly available Application Programming Interface (API) to build applications and publish them on the Android app market Hashimi, (2009).

The storage and transfer of sensitive information such as payment information using mobile devices, have directly led to the increasing danger associated with malware targeting these devices. Although their computing capabilities provide useful services to the users, they also open up serious security and privacy concern issues. Malware are malicious software that perform malicious actions such as information stealing, espionage and so on. Malware destroys valuable and sensitive information in infected devices same way it harms computers.

Mobile commerce (m-commerce) applications on the other hand, are deployed on smartphones and small wireless devices such as tablets for the exchange of goods and services. They have been found to be an easy tool in transacting businesses online and for the payment of goods and services have also been delivered through this mobile facility because of its high speed and anytime access. As an indispensable part of business environment, payment with paper currency and face-to-face method of payment for goods and services has existed for centuries, but with the rapid development of technology via internet environment and mobile devices, payment has been persistently changing from traditional methods to the ones faster and more convenient. Mobile payment refers to payment services operated under financial regulation and performed via a mobile device.

SMS-based mobile transactions between customers and merchants transmitted via network operators have made the payment of goods and services easier in recent time. All the customer needs to do is text a premium SMS to the service provider which includes the cost of items or services purchased. In most cases, the completed transactions take less than a minute and customers do not have to worry about getting their bank or credit cards details exposed. This is why over the years SMS payments have been considered safe, easy and secure (Roberts, 2015). The vulnerabilities of the messaging design in Android OS have been abused by malware Apps which target the OS. These Apps present itself as a useful regular SMS messaging application or other applications and use its basic permissions to Send/Receive short messages. Most permissions post by some Apps can put users at risk as some group of Android users do not understand the reason for and the difference between the various permissions requested by the application. Because of its worldwide acceptability for users to transfer credits/units through SMS, this application indirectly abuses these services to transfer credits from users illegally (Hamandi, 2013).

In an SMS-based payment system, SMS confirmation messages are sent by the service provider to confirm the authentication and authorization of transactions. This method has recently been under heavy attack especially by smartphones Trojans, such as, SMSTrojans (OpFake, Boxer, Xafecopy, and BeanBot). They capable of sending messages to premium-rate phone numbers, intercept and delete incoming text messages to hide alerts from mobile network operators without the user's knowledge and approval thereby causing financial damages to users. Variety of security risks will emerge in mobile payment because of its huge potential market.

The diagram in figure 1.1 shows the behavior of an SMSTrojan system. As can be seen, the user starts the malicious mobile payment application and an SMS is sent. Consequently, an attack by this type of malware is designed exclusively to target mobile devices thereby generates profit for the attacker (Goujon, 2016).

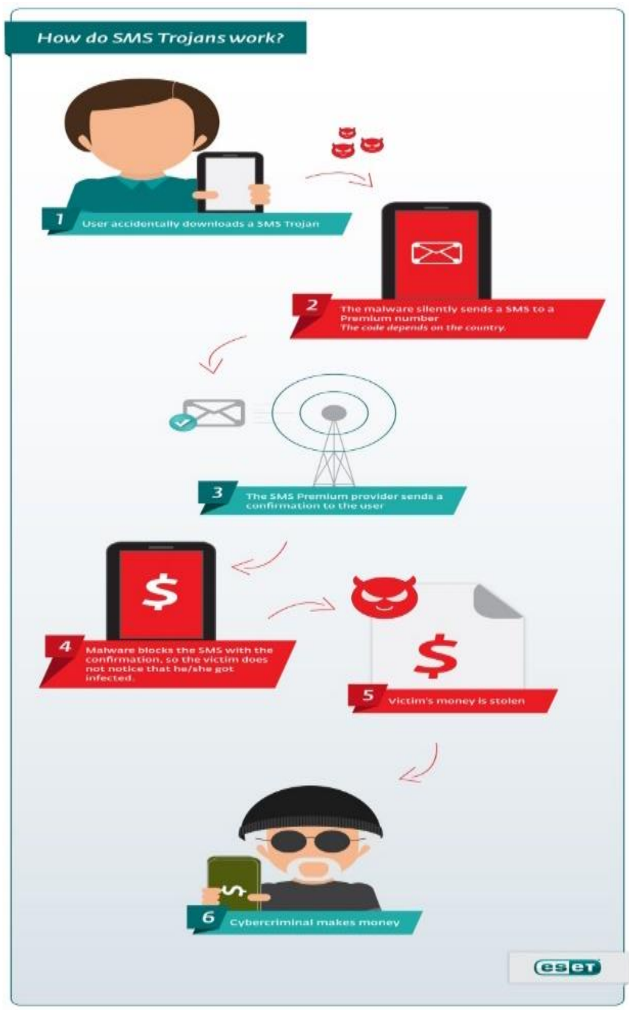


Figure 1.1: How SMSTrojan Malware operates (Goujon, 2016).

Machine learning techniques on the other hand, have witnessed unprecedented acceptance and usage in different sphere of computing and its application on different platforms. Most importantly the role it plays in cyber security has resulted in diverse security application especially in mobile payment system. Machine learning techniques has the ability to detect existing and new or unknown malware, study their behavior, intent and proffer ways of curbing them. Imran, (2016) stress that existing malware can be detected through the development of algorithms that are able to learn and predict their behavior as presented in Wenjia, (2015) using Support Vector Machine (SVM).

SVM is one of the machine learning classifiers receiving the most attention currently, and its various applications are being introduced because of its high performance. SVM can solve problem of classifying both linear and non-linear data and this compared to other machine learning classifiers can perform better in the aspect of complexity or accuracy of analysis. This paper employs SVM learning classifier capable of detecting malware with these malicious activities and curbing its wide spread on Android platform.

2. LITERATURE REVIW AND RELATED WORK

Burguera et al., (2011) developed Crowdriod, a behavioural-based detection system to detect malicious application by monitoring system calls using dynamic analysis that collects logfiles traced from an unlimited number of real users based on crowd sourcing and netork-based detection system traced to remote server. The system detected anomalously behaving application (malware Trojan) using k-means clustering algorithm. Although the system was capable of detecting 100% of self-written Trojan malware, the system still required multiple users to use the application.

A behavioural malware detection framework for android devices was developed in Shabtai *et al.* (2012). Andromaly employed a Host-based malware detection system that continuously monitors various features and events extracted based on analysis of various system metrics obtained from the mobile device. Evaluation of several combinations of anomaly detection algorithms (k-means, decision tree, logistics regression, histogram Bayesian networks and naïve Bayes), feature selection method and the number of top features were considered in order to find the combination that yields the best performance in detecting new malware on Android. Decision tree classifier yielded a good performance when trained features are trained on the same device while, naïve Bayes classifier performed best when trained on different device. Battery drainage deficiency was however recorded.

Faruki et al., (2013) developed 'AndroSimilar' to detect unknown variants of existing malwares that are usually generated using repackaging and code obfuscation techniques. Signature based malware detection technique was employed inconjuction with code obfuscation technique in order to change the signature of the code, thereafter, fuzzy hashing approach was applied to verify the robustness of signature. Results shows AndroSimilar was able to detect repackaged and obfuscated code. However, the system only could detect known malware variant and high level of false positive was recorded as a result of limited AndroSimilar signature database.

Abah *et al*., (2015) employed the dynamic analysis approach in detecting malware (SMS and calls) on Android platform. The system employ normality model in generating the behavioral pattern of the applications. It monitors and extracts device behavior via the list of monitored features such as In/Out SMSs, In/Out Calls, Device Status and Running Applications/Processes on Android applications data at the application layer and using these data to detect malware infections using a supervised machine learning approach (K-nearest) and WEKA tools. Results reveals high performance in the detection rate of the classifier with accuracy of 93.75%, low error rate of 6.25% and low false positive rate with ability of detecting real Android malware.

Sharma (2015) developed a Behavioral-based TrojanSMS Detection System for Android Mobile Device that is capable of sending SMS messages from infected device thereby causing financial loss to the user. The framework AndroLogSec (ALS) employed a dynamic behavior based detection system which monitors the behavior of an app on the modified Android source code and its detection system then detect the malware using the captured behavior and predefined malicious pattern.

AndroLogSec captures the dynamic behavior such as hooks in the Kernel to capture the system calls, changes in the framework to capture the API call or Inter Process Communication (IPC), changes in the Application to capture the behavior or combination of two or more behavioral technique stated early. Created some malicious patterns such as App is on screen and sends more than one SMS on a single click/touch event and App is not on screen and send one or more than one SMS. Naive Bayes classifier was used for training and the implementation was provided by scikit-learn. 5-fold cross validation was used for tuning the parameters. The overall accuracy of the classifier on the test set was 83.33% on the limited test set. There were no available malware samples for the detection so they developed their own Trojan SMS for the detection.

3. Methodology

SVM machine learning classifier is employed for detecting malware that sends SMS messages without requiring user's authorization. The training dataset is collected from a public malware repository, Urcuqui and Navarro (2016). Feature selection is performed to remove redundant permissions (features) using Singular Value Decomposition (SVD) technique. The relevant features are stored in a database using MySQL database model. Python programming language, Google TensorFlow and Android studio is employed in system implementation. Figure 3.1 represent the conceptual framework for SMSTrojan detection system.

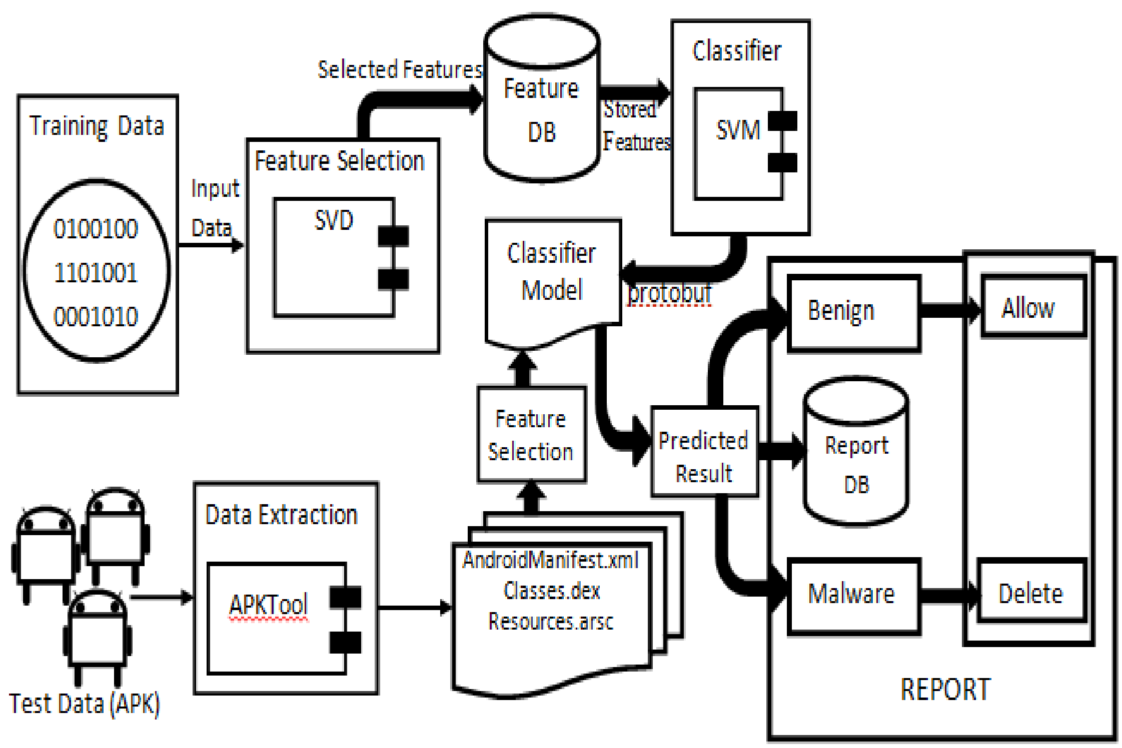


Figure 3.1: Conceptual Framework for SMSTDroid Detection System

Journal, Advances in Mathematical & Computational Sciences
Vol. 8 No. 3, September, 2020
www.mathematics-computationaljournal.info

3.1 Feature Extraction

The training and test datasets is a collection of permissions used by android apps to utilize system resource (e.g. SMS, Internet, Phone state, location) before being installed. The collected dataset is extracted permissions being frequently used by malicious applications. The features employed in this work were extracted from Urcuqui and Navarro database repository. The dataset used consist of 800 data points. The data points (0 = permission not requested and 1 = permission requested) generated into feature vector are divided as malware (-1) and benign (+1) as presented in Table 3.1.

Table 3.1: Feature Set

S/N	WRITE_CONTACTS	READ_CONTACTS	RECEIVE_SMS	SEND_SMS	RECEIVE_BOOT_COMPLETED	READ_SMS	WRITE_SMS	WRITE_EXTERNAL_STORAGE	READ_PHONE_STATE	TROJAN
1	1	0	1	1	1	1	0	1	0	-1
2	1	1	1	0	1	1	0	1	0	+1
3	0	1	1	1	1	1	1	0	1	-1
4	1	0	0	0	1	0	1	1	1	+1
5	0	1	1	1	1	1	0	1	1	-1
6	1	0	1	1	1	1	0	1	1	+1
7	0	0	1	1	1	1	0	1	1	-1
8	1	0	1	0	1	0	1	1	1	+1
9	1	1	1	1	1	1	0	0	0	-1
10	1	1	0	1	1	0	0	1	0	+1
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
797	1	1	1	1	1	1	1	1	1	-1
798	1	1	0	1	0	1	0	0	1	+1
799	1	0	1	1	1	1	0	0	1	-1
800	1	0	0	1	1	1	1	1	0	+1

The selection of the major relevant permissions was done using dimensionality reduction technique (SVD). The label column contains the interpretation of each data point interpreted as -1 meaning “SMSTrojan” and +1 meaning “Benign” application.

3.2 Feature Selection

Feature selection is used for removing redundant and irrelevant features to improve the accuracy of the predication. In this paper, Dimensionality Reduction Technique is employed to find a lower-dimensional representation of a dataset such that as much information as possible about the original data is preserved. Singular Value Decomposition (SVD) is one of the several techniques that can be used to reduce the dimensionality. The SVD model is presented as

$A_{mn} = U_{mm}S_{mn}V_{nn}^{\tau}$ (Equation 3.1)

where;

U - is an orthogonal matrix U,
S - is a diagonal matrix, and
V - is the transpose of an orthogonal matrix

To obtain

$$A_{mn} = U_{mm}S_{mn}V_{nn}^{\tau} = \begin{bmatrix} 0.7071 & 0.7071 \\ 0.7071 & -0.7071 \end{bmatrix} \begin{bmatrix} \sqrt{3} & 0 & 0 \\ 0 & \sqrt{1} & 0 \end{bmatrix} \begin{bmatrix} -0.5774 & -0.5774 & -0.5774 \\ 0.4081 & -0.8167 & 0.4081 \\ 0.4082 & -0.8165 & 0.4082 \end{bmatrix} =$$

$$\begin{bmatrix} 1.2247 & 0.7071 & 0 \\ 1.2247 & -0.7071 & 0 \end{bmatrix} \begin{bmatrix} -0.5774 & -0.5774 & -0.5774 \\ 0.4081 & -0.8167 & 0.4081 \\ 0.4082 & -0.8165 & 0.4082 \end{bmatrix} = \begin{bmatrix} -0.4185 & -1.2846 & -0.4185 \\ -0.9957 & -0.1296 & -0.9957 \end{bmatrix}$$

3.3 Classifier

Support Vector Machine (SVM) was employed to classify the android application (test data) into SMSTrojan. The input data are formulated as seen below

$(x_1, y_1\ ) \ldots (x_n\ \ y_n)$ Equation 3.2

where:

x is the feature set
y is the label
x_i *is the feature set*

$x_i =\ x_i^1, x_i^2, \ldots\ldots x_i^d$ (Equation 3.3)

where:

x_i^j: is a real value

$y_i = \{-1, +1\}$ (Equation 3.4)

with -1 representing “SMSTrojan” and +1 representing “Benign”

Radial Basis Function (RBF) Kernel is used to map the non-separable training data from input space to feature space in order to find an optimized hyperplane that correctly segregates the data. The RBF kernel is presented in Equation 3.5;

$K(\vec{x_i}, \vec{x_j}) = \emptyset(\vec{x_i})^T \emptyset(\vec{x_j})$ (Equation 3.5)

$K(\vec{x_i}, \vec{x_j}) = \exp(-\gamma \, ||x_i - x_j||^2)$ (Equation 3.6)

where:
$\gamma: \frac{1}{2\sigma^2} > 0$

x_i: Support vector points

x_j: Feature vector points in the transformed space

$K(\vec{x_i}, \vec{x_j})$: Kernel function

The kernel function calculates the dot product of the mapped data points in the transformed feature space. The optimal hyperplane that segregates between the two classes ("SMSTrojan", "Benign") is found using equation 3.6;

$w^T.x + b = \sum_{i=1}^{l} \alpha_i \, y_i \, \emptyset(\vec{x_i})^T \emptyset(\vec{x_j}) + b = 0$ (3.7)

The classification frontiers are found by the following equation
$w \, \emptyset(x) + b = -1$: for points labeled as "SMSTrojan"
$w \, \emptyset(x) + b = +1$: for points labeled as "Benign"

The optimal weight vector (w) is given by;
$\vec{w} = \sum_{i=1}^{l} \alpha_i \, y_i \, \emptyset(\vec{x_i})$ (Equation 3.8)

This work used the dual formulation of SVM algorithm. This formulation which is presented as a maximization problem over α is shown in Equation 3.8;
$\max \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \, \alpha_j \, y_i \, y_j \, \emptyset(\vec{x_i})^T \emptyset(\vec{x_j})$ (Equation 3.9)
Subject to:
$0 < \alpha < C$
and $\sum_{i=1}^{l} \alpha_i \, y_i = 0$
Where:
α_i: is the weight vector
y: is the label vector
$\emptyset(\vec{x_i})^T \emptyset(\vec{x_j})$: is the kernel function
C: is the intercept

Journal, Advances in Mathematical & Computational Sciences
Vol. 8 No. 3 September, 2020
www.mathematics-computationaljournal.info

The decision function g used in making prediction is given as;

$$g(\vec{x}) = sgn(\vec{w}^T\vec{x} + b)$$
$$\Rightarrow sgn\left(\sum_{i=1}^{l} \alpha_i \, y_i \, \emptyset(\vec{x_i})^T \emptyset(\vec{x_j}) + b\right) \quad \text{(Equation 3.10)}$$

Where:
$g(\vec{x})$: is the predicted label
sgn: is the sign of $(\vec{w}^T\vec{x} + b)$ (i.e. -1 or +1)
α_i: is the weight vector

4. RESULT AND DISCUSSION

4.1 SVM Classifier Result

This section shows the result of the SVM classifier with an optimized hyperplane that correctly segregate between the two classes. From the SVD dataset transformation, 600 datasets were employed for training and 200 datasets for validation of the trained model. Figure 4.1 shows the result of the 600 trained datasets using Support Vector Machine algorithm.

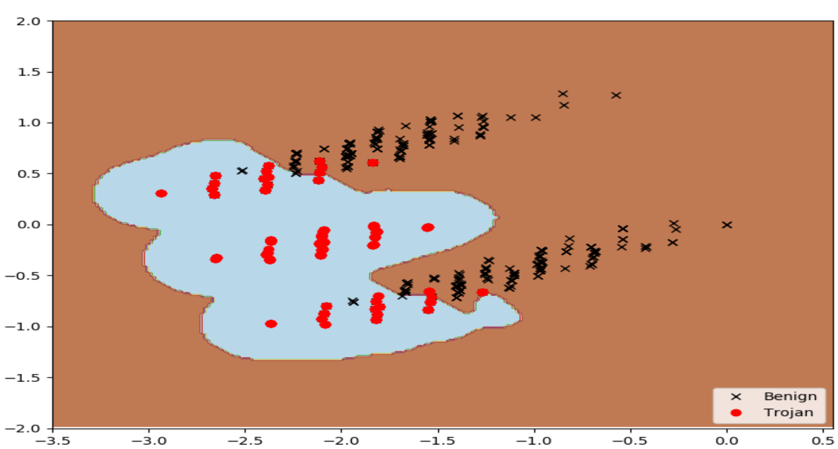


Figure 4.1: Plot of SVM Classifier

4.2 Metrics for Performance Evaluation of the SVM Classifier

The performance evaluation metrics used to evaluate the system performance are True Positive Rate (TPR), False Positive Rate (FPR), Recall, Precision and Accuracy and this focuses on the predictive capability of a model. Figure 4.2 is the screenshot that shows the calculated values of the various performance metrics using the formula stated.

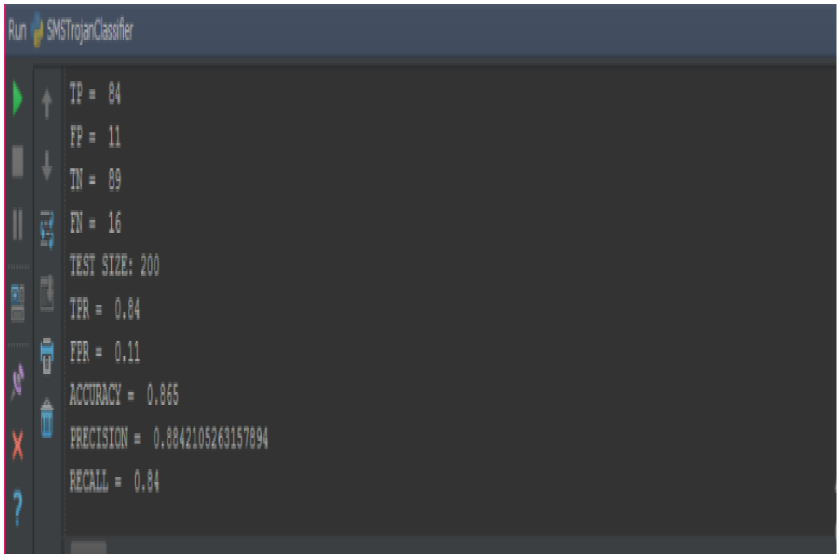


Figure 4.2: Value of Performance Metrics

Table 4.2

TP	FP	TN	FN	TEXT SIZE	TPR	FPR	ACCURACY	PRECISSION	RECALL
84	11	89	16	200	0.84	0.11	0.865	0.884	0.84

The validation result is represented as a pie chart shown in Figure 4.3. From the 200 datasets used for validation, 26 were wrongly classified represented as 13% and 174 were correctly classified represented as 87%. This shows effectiveness in our model with minimal misclassified instances.

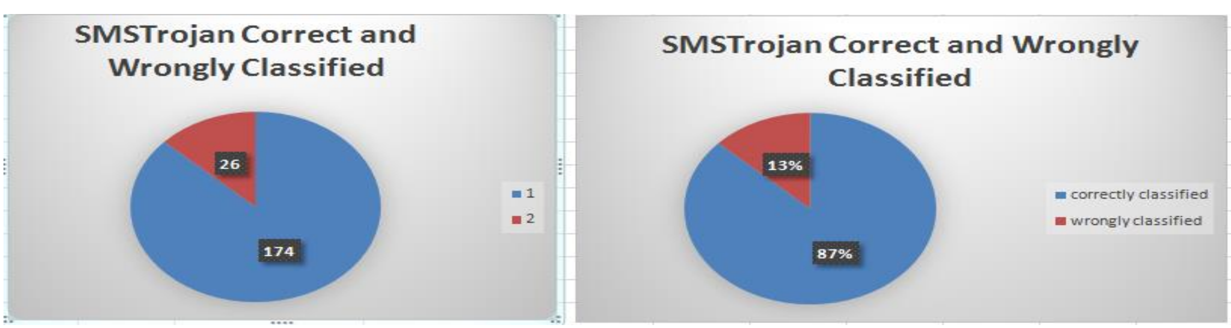


Figure 4.3: Pie Chart of the Result of Validation dataset

4.3 SMSTDroid Detection System Implementation

The implementation of the SMSTDroid detection system aims at capturing the possible malicious behaviors and other unique features of an individual application to be analyzed. The SMSTDroid application is implemented for Android Operating System written in Java. An Android studio along with Android Software Development Kit (SDK) and the Java programming environment will provide the necessary tools to compile the Java source code and generate the APK file that will run on the devices.

We obtained the following results from our implemented SMSTDriod system:

1. The User Interface (UI) of the SMSTDroid system in Figure 4.17 shows the entry point of the system and it loads the system's resources by monitoring and collecting information from the applications running on the device. The UI displays importance details of the system such as Application name (SMSTDROID), developer's name, registration number and department.

Figure 4.4: Screen shot of the SMSTDROID Application

2. During the monitoring process, the system displays the scanning activity that displays all the actions that takes place during the scanning process with the current applications being scanned and a progress indicator shows a successful scanned process as shown in Figure 4.5.

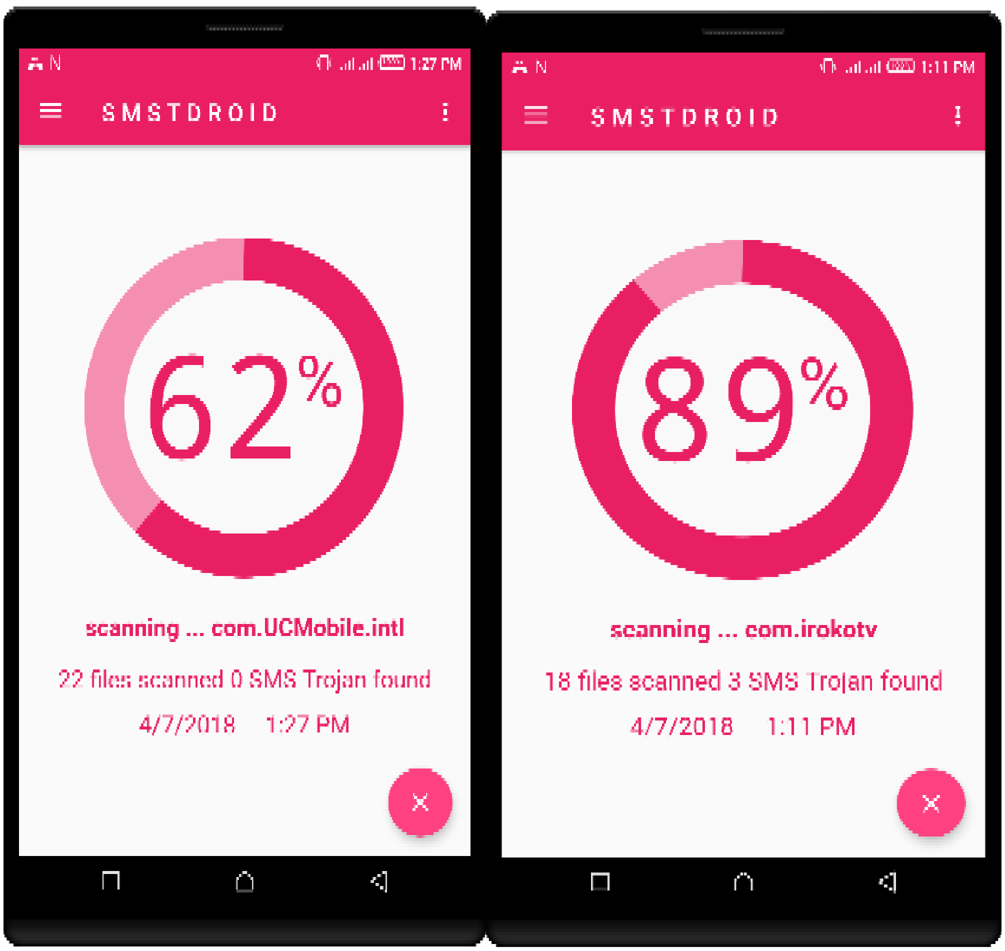
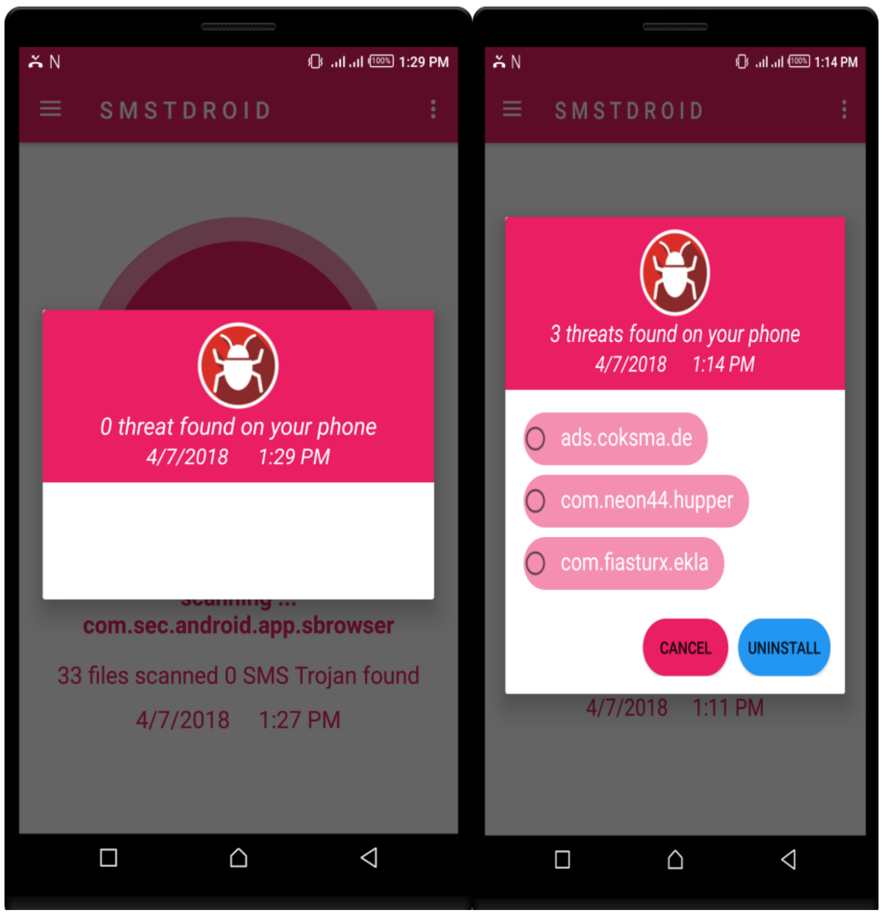


Figure 4.5: Screen shot of the SMSTDROID Application scanning process

Journal, Advances in Mathematical & Computational Sciences
Vol. 8 No. 3, September, 2020
www.mathematics-computationaljournal.info

3. Figure 4.6 shows the screen shots of SMSTDroid after monitoring and detection were carried out on feature vector without any malicious instances and then on feature vector with malicious instances and a notification dialog immediately display the results obtained from the scanning process and the user gets an opportunity to cancel or uninstall a suspected application that is malicious.



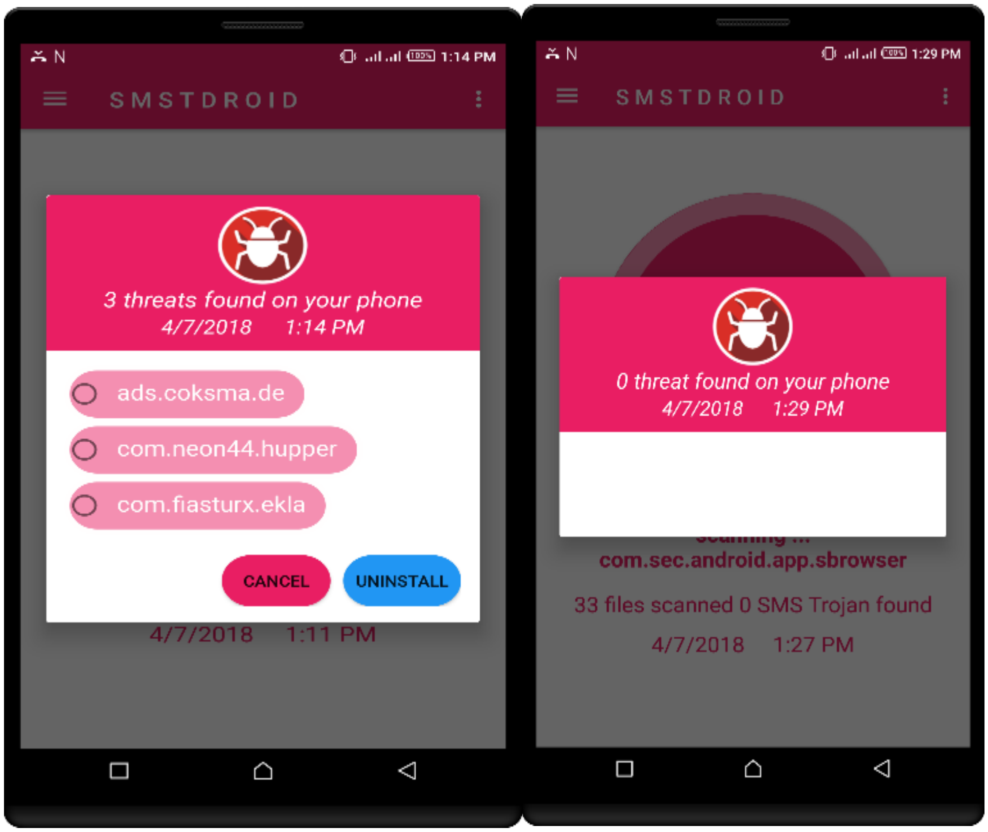


Figure 4.19: Screen shots of Benign and Maliciously classified cases

1. Figure 4.20 shows the screen shot of the SMSTDroid Application menu with its Application's global functions. These functions include:
    1. Report used to view the scanning results,
    2. Log used to view system activities,
    3. Settings used to modify system's behaviors,
    4. Feedback used to send suggestions to developer

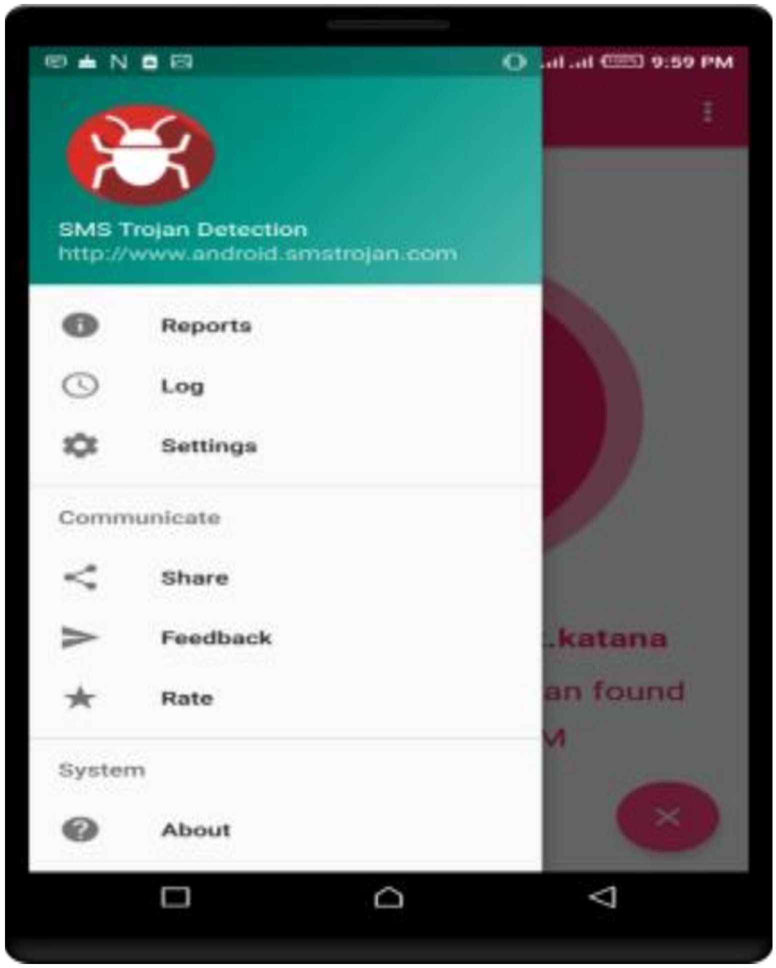


Figure 4.20: Screen shot of the SMSTDROID Application Menu

5. CONCLUSION

Smartphones are becoming popular in terms of power and communication. Due to its multi-functionality, security threats especially SMS malware have emerged that affect the messaging design of these smart devices. In this work, an SVM learning classifier is used to implement an SMS-based Trojan malware detection system for mobile payment system on Android platform. The SVM model is effectively trained and used on the test dataset to predict instances as either malicious or benign as summarized in Table 4.6. The SVM model performance yields promising results of 0.865 accuracy rate with an error rate of 0.135. A low false alarm rate of 0.11 is predicted by the model which implies that the number of misclassified malicious instances as benign is minimal.

REFERENCES

1. Gartner Press Release (2016). *Market Share: Mobile phones 1Q16 units.* Egham, UK. https://www.gartner.com/newsroom/id/3859963, Retrieved on 2nd October 2017.
2. Nuremberg (2019). Glbal Smartphone Sales reached $22 billin in 2018. Available at: gfk.com
3. Mongardini J. and Radzikowski A. (2020). Global Smartphone Sales May Have Peaked: What's Next? Available at: imf.org
4. Hashimi S.Y. and Komatineni S. (2009). *Pro Android.* Verlag, New York, Apress Inc., ISBN-13(pbk): 978-1-4302-1596-7, pp.4-15.
5. Goujon, A., and Ramos P. (2016). *Android_Trojan-SMS*.
6. http://www.virus-radar.com/en/Android_Trojan SMS/description. Retrieved on 8th February, 2018.
7. Imran, M. (2016). *Evaluation of Hidden Markov Model for Malware Behavioral Classification.* Ph.D. Thesis, Department of Computer Science, Capital University of Science and Technology, Islamabad, Pakistan.
8. Burguera, I., Zurutuza, U. and Nadjm-Tehrani, S. (2011). *Crowdroid: Behavior-Based Malware Detection System for Android.* In: Proceedings of the 1st ACM workshop on Security and Privacy in smartphones and mobile devices, Chicago, USA, pp.15-26.
9. Shabtai, A., Kanonov U., Elovici Y., Glezer C. and Weiss Y. (2012). *Andromaly: A Behavioral Malware Detection Framework for Android Devices.* Journal of Intelligent Information Systems, 38(1):161-190.
10. Faruki, P., Ganmoor, V., Laxmi V., Gaur M. S., and Bharmal A. (2013). *Androsimilar: Robust Statistical Feature Signature for Android Malware Detection.* In: Proceedings of the 6th International Conference on Security of Information and Networks, Aksaray, Turkey, pp.152-159.
11. Abah, J., Waziri, O. V., Abdullahi, M. B., Arthur, U. M. and Adewale, O. S., (2015). *A Machine Learning Approach to Anomaly-based Detection on Android Platforms*. International Journals of Network Security and its Applications (IJNSA), 7(6):5-10.
12. Sharma, R. (2015). *Behavioral Based Trojan-SMS Malware Detection in Android.* M. Tech., Thesis, Department of Computer Science and Engineering, Indian Institute of Technology, Bombay, India.