Malware Detection Based on Permissions on Android Platform Using Data Mining

Tawfiq S. Barhoom[1], Mohammed I. Nasman[2]

[1]*Islamic University - Gaza, tbarhoom@iugaza.edu.ps*

[2]*Islamic University - Gaza, mnasman@gmail.com*

Abstract— With the spreading of smart mobile devices to nearly every person, Android Operating System is dominating the mobile's operating systems.

Due to the weak policy of submitting application to Google Play store, attackers developed malware to attack the users of the Android operating system with malware application or by including malicious code into applications. Researchers have been done in this area, but solutions required installing the applications to monitor the malware behavior, or by taking actions after installing the application .We proposed a new method using Data Mining to detect newly and unknown malware using the applications' permissions as base features. In order to create binary dataset we collected up to "103" benign and malware android app samples, the dataset consist of five different features collected based on different number of attributes and conditions. Different evaluation measure used to evaluate the proposed method, the results show that we achieved 96.74% with f-measure and 0.993 with area under the ROC curve..

Index Terms— Malware, Android, Data mining, Permissions, APK, Classifications.

I.INTRODUCTION

With the repaid growing of Android application every day, there are growing threats for the mobile users by installing more malwares without ability to detect them before installing the applications to the user device.

Malware name came from "Malicious Software", its software designed to secretly access a system without the owner's device knowledge. Malware can effects mobile resources, or just make the devices not responding to the users, it may go to dangerous behaviors like steal private information, without the user notice any harmful action[1]. Malware has different types, PCs and Mobiles has the same types, which can be listed on different categories such as : Adware, Bots , Rootkit, Spyware, Trojan horse or a "Trojan," , Virus, Worms.

According to data from the International Data Corporation (IDC) the worldwide smart phone market grew 27.2% year over year in the second quarter of 2014, just over a third of a billion shipments at 335 million units.2014 promises to close at nearly 1.3 billion shipments, with Android taking the lion's share, spread across over 180 tracked vendors[2]. Market research firm Strategy Analytics has given the numbers for the second quarter of 2014 that estimate the market share of Android platform's on the global market has reached 84.6 percent.[3]

For the mobile devices that use Android as its platform, the official way to install the applications is the Google play store[4], which serve as repository of the application developed for Android, and it installed by default with all the Android devices. The current reviewing process for the applications submitted by developers to Google Play store took only two hours[5], compared to process for Apple App Store takes 6 days[5].

Google may phase out the discovered malware but after it's spreading, for example: More than 50 applications on Google's Android Market have been discovered to be infected with malware called "Droid Dream" which can compromise personal data by taking over the user's device, and have been "suspended" from the store[6]. Currently mobile malware detection tools uses pattern recognition to identify the malware, but it fails to distinguish the threats. Android gives accessing to the device's resources (such as writing files, accessing the internet, locations, SMS, etc.), with permissions system, which they defined on each Android Application Package (APK) in special file called "AndroidManifest.xml".

Any application needs to access any of these resources will define the resources required on "AndroidManifest.xml" on development time, after the application compiled and uploaded to Google play Store, it will show to the users the permission required for installing the application. But with lack to understanding and knowledge for most of users, they can install the application that has access to special resource and it may be has a harmful use.

According to above, the need to a new method to recognize the malware applications before installed by the users is important to prevent the malware attack their mobile resources and Data .This paper focus on new method for

detecting Malwares based on permissions required by the applications, using classification techniques to detect malware apps from benign.

II.RELATED WORK

Many researches used different approaches to detect the malware, some of the methods require process on the mobile devices, and other methods do the processing on the cloud from the data collected on the mobile device:

Sanz et al[7], presented detection method using string analysis that will get the strings from android application by disassembling the Android application and then extract the strings in const-string and using machine learning to training the dataset and assign category (malware or goodware).The problem with method, that developers of malware may using non English languages in const-strings will not make them detectable by this method, also if the developer of the malware application encrypt the strings, they will not be detectable in this method too.

Burguera and Zurutuza[8], have developed a framework for detecting malware on android platform, the framework consist of multiple components: Data acquisition which using application developed "Crowdroid" is small application installed from Google Play store, and it will monitor Linux calls on the device, and compare it from same application that downloaded from other resources, then it may detected if the application is modified with some malware code, the other component is Data manipulation: this component will manage and parse the data collected from the android users, and Malware analysis and detection component: which is used to analyzing and clustering the feature vectors extracted from the other components.

The method developed consist of several tools on client and server side, the main problem with this method that if the malware application submitted to Google Play store and has no other resources, it will not detect the application is malware. Yerima et al[9], proposed and evaluated a new approach for detecting Android malware by reverse engineering the Android applications using APK Analyzer, and building the dataset from set of 58 properties from API call, Commands and Permissions, then used a Bayesian based classifier for learning and detection stages. The result of study showed the proposed method has better detection rates then signature based anti-virus, but the method require disassembly of application and then extracting the used features which may not suitable as preventable method.

Cheng et al[10], has presented a collaborative virus detection and alert system for smart phones (SmartSiren), they used behavioral analysis of smart phone viruses by ontology, the certainty factor function (CF function) generation by the certainty factor theory and the reasoning process of detecting viruses by a FPN model. They developed mobile malware detection system (MMDS), which will filter the files received by SMS or MMS by extracting their behaviors and determine the danger level and if the users have confirmed them danger of these files, the system will reject the files sent by the SMS or MMS. The presented method require an application to be installed on the smart phone (MMDS), and also require an interactive from the users to confirm the danger of the files, novice users will have hard time determining if the received SMS or MMS has danger file or not, especially if that received from known numbers. Koundel et al[11], proposed a method to build a dataset from installed application on user mobile phone, the method using an application that will be installed on user mobile and send list of the applications installed, and the permissions and applications battery's usage, then sent to server to as csv file, then serve will parse the csv file and build it into the dataset.

The downside of this method it's require an application to be installed to gather the data from the end user mobile, also the application may itself drain the battery, which is another downside of this method. Liu et al[12], proposed general Malware detection method called Virus Meter, it's monitor the usage of battery power on mobile devices, and compare it to pre-defined power consumption model to identify the abnormal usages of the battery power, using the OS Api it will calculate how much power used by the running services, and compared to the pre-defined model. The proposed model monitor only the power usage of the system to determine if there's a malware on the mobile device or not, it may gave misleading alarm based on s normal services may require more power for various reasons such as background updating, or downloading the data.

Sahs and Khan[13], in this paper the authors used an open source application called "Androguard" to extract features of the APK file and used Scikit-learn framework to train vector machine to generate as much as positive marked as negative if there's enough differences from the training data. This method treat all applications as benign except if it's sufficiently different from training data, so this may mark malware application as benign because there's not previously added in the training dataset.

Jacobsson et al[14], Built two models "bag-of-words" and "meta EULA model" to find spywares, they collected more than 1000 (900 clean, 100 Bad) of "End User License Agreements (EULAs)" and they apply the model with multiple classifiers such as: Naïve Bayes, Decision Stump, J48, Etc , and results support their hypothesis that EULAs can be used as a basis for classifying the corresponding software as good or bad.

This method will not work if the spyware authors start to copy the good applications EULA and use them with same text. Shaban[15], has built a model to detect the spyware using data mining for windows portables files (PE), the researcher collected many windows PE that include benign and spyware executable files, then exported the API calls and put them on categories, then apply data mining classification for detecting the spyware.

The proposed model in this study require the files to be saved first, after that the file need to be analyzed to extract the API calls, we need to find a way to find if the file is malware before install it.

III. DATA MINING CLASSIFICATIONS METHODS

We evaluating a variety of classification methods such as:

k-Nearest Neighbor(kNN), Naïve Bayes, Support Vector Machine (SVM)and Decision Tree, we used these classifications methods with different feature sets.

Performance Evaluation

1. Coincidence Matrix

For the classifications problems the main source of performance measurement is the coincidence matrix.
we can calculate most commonly used metrics equations from coincidence matrix as shown in eq (1), eq (2), eq(3), eq (4), eq (5).

$$\text{True Positive Rate} = \frac{TP}{TP+FN} \quad (1)$$

$$\text{True Negative Rate} = \frac{TN}{TN+FP} \quad (2)$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (4)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (5)$$

2. Accuracy: is the percentage of true results (true positives or true negatives) between the total number of cases examined [18].

$$\text{Accuracy} = \frac{No\,of\,TP + No\,of\,TN}{No\,of\,TP+FP+FN+TN} \quad (6)$$

3. Precision: is the correctly retrieved instances of query [19].

$$\text{Precision} = \frac{|(relevant\ dcouments) \cap (\ retrieved\ documents\ )|}{|(retrieved\ documents\ )|} \quad (7)$$

4. Recall: is part of the documents that relevant to the query that have been successfully retrieved[19].

$$\text{Recall} = \frac{|(\text{relevant dcouments}) (\ \text{retrieved documents}\ )|}{|(\text{relevant documents}\ )|} \quad (8)$$

5. AUC

Receiver operating characteristic is created by comparing the true positive rate (TPR) against the false positive rate (FPR) at various sill settings. The ROC recently introduced to evaluate ranking performance of machine learning algorithms[20].The AUC combine all of the features of ROC into single value, by calculating the area of inclination shape below the ROC, the closer ROC get into optimal point of prediction, the AUC gets closer to one[21].

$$FPR(\theta) < FPR(\theta') \rightarrow \theta > \theta' \rightarrow TPR(\theta) \leq TPR(\theta') \quad (9)$$

6. F-Measure:

F-Measure considered as weighted average between precision and recall, it's calculated as see in eq (1).

$$F1 = 2.\frac{precision.recall}{precision + recall} \quad (10)$$

7. Cross Validation

In k-fold cross-validation, initial data are indiscriminately divided into k reciprocally exclusive subsets or "folds", D1, D2, ..., DK, each one is approximately same size, training and testing done k times, in loop i , division Di, is set for test set, and the other divisions are used to train the model, and then for other Di, until DK..

8. Identification methods for the malware:

Mainly the malware detection techniques fall into these categories:

- **Signature based detection:** It's search for sequence of unique bytes that defined the malware, and compare it to the database of other malware data, most of Anti-Malware use this technique [22].

- **Behaviors based detection**: By monitoring many factors of the malware such as the source, target and other statistical properties, then evaluating the damage of the system under controlled environment using dynamic behaviors.

V. METHODOLOGY AND EXPERIMENTS

The method will work as shown in figure 1, when new application need to be downloaded, we read permission first, then after extracted them we will applying the classifier to the extracted data to find if the application is malware or not.

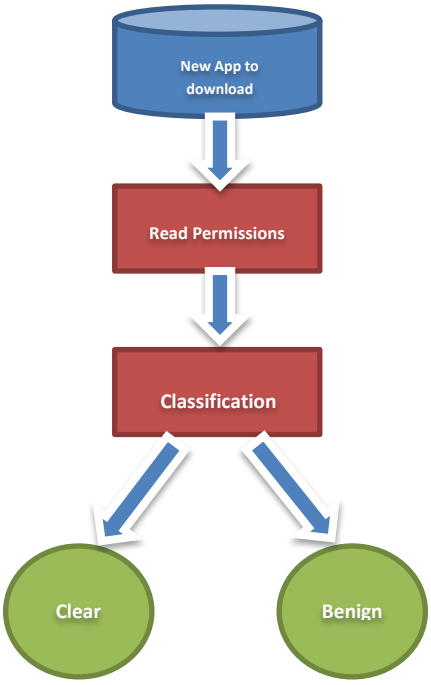

Figure 1 – overview of the method

Our method will start by collecting data to build the

Feature sets	Weight	Number of Attributes
Feature set 1	0.1	38 regular 1 special (from attribute 16 to 54 as listed on table 4.1)
Feature set 2	0.2	26 regular 1 special (from attribute 29 to 54 as listed on table 4.1)
Feature set 3	.03	15 regular 1 special (from attribute 40 to 54 as listed on table 4.1)
Feature set 4 (*dangerous permissions*)	No Weight	24 regular 1 special
Feature set 5 (extended *dangerous permissions*)	No Weight	27 regular 1 special

dataset, then finding an appreciate classifier for our method, and finally we will evaluate and test the method

1. Collect the data:

At first, we collect the benign and malware applications from different sources.

A. Benign Applications

The benign Applications has been downloaded from Google Play store, and due to Google policy it doesn't allow downloading the APK files directly from their website, but users can install them directly from the Play Store application on their Mobile device, also Android mobile phone doesn't allow to extract the APK files because they are hidden with the system files, so we used "APK Downloader"[23] website to download the APK files, it's simulate the mobile devices as it act as mobile, then offering the APK to be downloaded from the website directly to our PCs. The downloaded files with APK Downloader has been verified from virus using "Virus Total" website, which verify the uploaded files with 53 Anti-Virus to make sure the applications has not been infected by any Malware.

B. Malware Applications:

The malware dataset has been downloaded from "Free Range Security"[16], it's containing 189 malware Applications.

2. Extract Permissions

Then we extract the permissions from APK files. The Android Asset Packaging tool and the Read Permissions tool built for automate this work, and to export the extract permissions into one file, after we cleaned the data and built our dataset from five different feature set, based on weight for attributes and from dangerous permission listed provide by Google.

4. Building up Dataset:

We use the three features sets by weight and the one that contain Google's dangerous permissions as fourth Feature, the last feature set was the attributes used more in malware than benign and not listed in dangerous attributes list, the final dataset described in table 1.

Table 1
Weighted Feature sets

The experimental environment that used for all the experiments was laptop with core i7 CPU, 500GB SSD with 16GB Ram. Software and Tools are used , **Rapid Miner 5, Microsoft Excel2010 , PSPad , Android SDK 4Tools , Delphi 2010 , 7-zip.**

4.Apply classification and Evaluation the method

After we prepared our dataset with 5 different feature sets, we applied the classification algorithms (**K-NN , Naïve bayes , SVM , Decision Tree**) . The settings set in the evolution phase for each classifier as following Figure 2

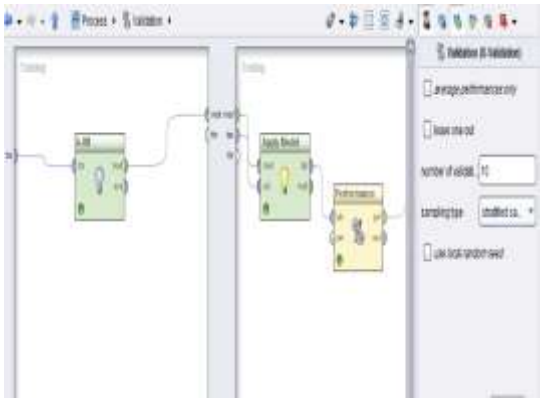

Figure 2. Rapid Miner with kNN and validation process

● **Experiment Scenarios 1(feature set with Weight > 0.1)**: the number of samples are 103, and number of attributes are38, the SVM classifier was a higher in both AUC & F-Measure, as shown in Table 2 and Figure 3

Table 2
Experimental Result with feature set 1

Classifier	*AUC*	*F-Measure*
K-NN	0.5	88.58%
Naïve Bayes	0.989	92.31%
SVM	**0.993**	**95.20%**
Decision Tree	0.756	86.39%

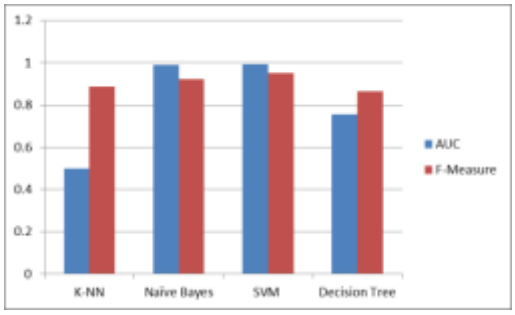

Figure 3. Experimental Results of feature set 1

● **Experiment Scenarios 2 (feature set with Weight > 0.2)**:

In this experiment the number of samples are 103 and number of attributes are 36 , here the Naïve Bayes classifier was a higher in both AUC & F-Measure, but SVM gave the same value with AUC as NB see Table 3 and Figure 4

Table 3
Experimental Result with feature set 2

Classifier	*AUC*	*F-Measure*
K-NN	0.5	88.89%
Naïve Bayes	**0.993**	**96.74%**
SVM	**0.993**	94.48%
Decision Tree	0.773	92.04%

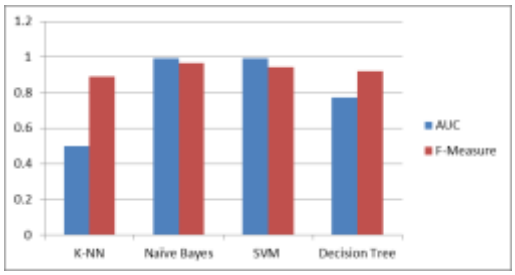

Figure 4. Experimental Result with feature set 2

● **Experiment Scenarios 3 (feature set with Weight > 0.3):**

We applied the 4 classifier to the dataset, the number of samples are 103, and number of attributes are 15, the Naïve Bayes classifier was a higher AUC and SVM gave the higher value with F-Measure, as shown in Table. 4 and Figure 5

Table 4
Experimental Result wit feature set 3

Classifier	*AUC*	*F-Measure*
K-NN	0.5	95.56%
Naïve Bayes	**0.988**	90.43%
SVM	0.985	**95.75%**
Decision Tree	0.766	92.17%

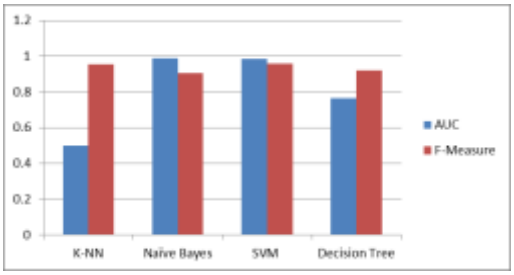

Figure 5. Experimental Result with feature set 3

● **Experiment Scenarios 4 (Dangerous Permissions):**

We applied the 4 classifier to the dataset, the number of samples are 103, and number of attributes are 15, the Naïve Bayes classifier was a higher F-Measure and SVM gave the higher value with AUC ,as shown in Table 5 and Figure 6 :

Table 5
Experimental Result with Dangerous permissions feature set 4

Classifier	*AUC*	*F-Measure*
K-NN	0.500	85.93%
Naïve Bayes	0.979	**92.85%**
SVM	**0.985**	89.98%
Decision Tree	0.908	91.59%

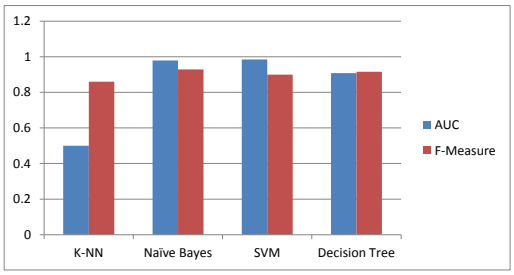

Figure 6. Experimental Result with Dangerous permissions feature set 4

● **Experiment Scenarios 5 (Dangerous Permissions 2):**

We applied the 4 classifier to the dataset, the number of samples are 103, and number of attributes are27, the k value of kNN was 1, the naïve Bayes used with Laplace correction is checked, both SVM and Decision tree used with default values set by RM, as shown in Table 6 and Figure 7:

Table 6
Experimental Result with Dangerous permissions feature set 5

Classifier	*AUC*	*F-Measure*
K-NN	0.500	88.49%
Naïve Bayes	0.983	**94.64%**
SVM	**0.986**	92.79%
Decision Tree	0.894	92.13%

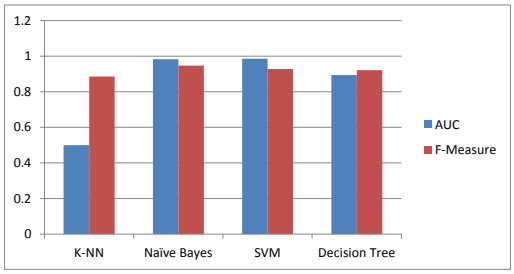

Figure7 . Experimental Result with Dangerous permissions feature set 5

From the experimental above, we notice the feature set 2 has the highest rates in the metrics we used for the evaluation (AUC and F-measure) as shown in Figure 8 .

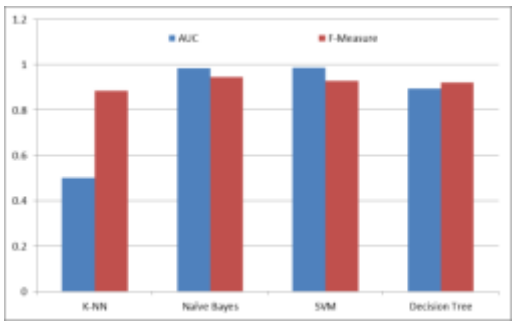

Figure 8. Experimental Result Summary

We achieved highest score in AUC (0.993) and F-Measure (96.74%) with Feature set 2 using Naïve Bayes classifier. Feature set 1 gave same high accuracy as feature set 2 in AUC (0.993) using Support Vector Machine SVM classifier. Feature set 5 (dangerous permissions with extended attributes) gave higher rates then feature set 4 (dangerous permissions specified by Google), We think these attributes should be consider by Google, to warn users about the dangerous effect of the attributes. In our experimental k-NN classifier has the worst performance in both AUC & F-measure in all feature set ,but Both Naïve Bayes and SVM, has the best performance in our experimental.

IV. Conclusion

In this paper we worked on building dataset from benign and malware Android Application.

Then we separate the database to five different feature sets based on attributes by weight and Google dangerous permissions.

After that we evaluated our method with Rapid, to find the most attributes that has effect for malware application.

Our results show that attributes with feature set 2 using Naïve Bayes classifier, gave us the most accurate result for detecting the malware.

Also we found that there are more attributes should be categories as dangerous attributes by Google, because in our experimental adding these three attributes gave us a better detection on all feature sets we used.

Our method aimed to detect the malware before installing them to the user's mobile device. However, with new thousands Applications added daily to Google play store, we need to find a better way to get the permissions of the applications without extracting them by downloading the APK first,

Currently, Google didn't provide any official API to access the information of the applications in Google play store information, some open source trying to achieve this, but may not work when Google change their protocol. Other options to reverse engineer the Google's API to find better way to get the permissions from the store, or by doing website scarping to gather the information from play store website.

References

[1]B. K. Addagada, "Intrusion Detection in Mobile Phone Systems Using Data Mining Techniques,," ed, 2010.

[2] International Data Corporation. (2014) IDC. [Online]. http://www.idc.com/prodserv/smartphone-os-market-share.jsp

[3] Strategy Analytics. (2014) [Online]. http://thenextweb.com/google/2014/07/31/android-reached-record-85-smartphone-market-share-q2-2014-report/

[4] Google. Play Store. [Online]. https://play.google.com/store

[5] Brendan Fitzgerald. (2014, Aug) appmakr.com. [Online]. http://www.appmakr.com/blog/how-long-app-approved/

[6] Charles Arthur. (2011, Mar) The Guardian. [Online]. http://www.theguardian.com/technology/blog/2011/mar/02/android-market-apps-malware

[7] Igor Santos, Javier Nieves, Carlos Laorden, Iñigo Alonso-Gonzalez, Pablo G. Bringas Borja Sanz, "MADS: Malicious Android application Detection through String analysis," Lecture Notes in Computer Science, pp. 178-191, 2013

[8] Urko Zurutuza Iker Burguera, "Crowdroid: behavior-based malware detection system for Android," ACM - Association for Computing Machinery, pp. 15-26, OCT 2011.

[9] S. Y. Yerima, S. Sezer, G. McWilliams, and I. Muttik, "A new android malware detection approach using bayesian classification," in Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on, 2013, pp. 121-128.

[9] Wikipedia. (2005) Wikipedia. [Online]. https://en.wikipedia.org/wiki/Android_(operating_system)#History

[10] Starsky H.Y. Wong, Hao Yang, Songwu Lu Jerry Cheng, "SmartSiren: Virus Detection and Alert for Smartphones," ACM (Association for Computing Machinery), pp. 258-271, June 2011.

[11] Suraj Ithape, Vishkha Khobaragae, Rajat Jain Deepak Koundel, "Malware Classfification using Navie Bayes Classifier for Android OS," The International Journal of Engineering and Science, vol. 3, no. 4, pp. 59-63, 2014

[12] Guanhua Yan, Xinwen Zhang, and Songqing Chen Lei Liu, "VirusMeter: Preventing Your Cellphone," Association for Computing Machinery, pp. 244 - 264, Oct 2009

[13] Justin Sahs and Latifur Khan, "A Machine Learning Approach to Android Malware Detection," in European Intelligence and Security Informatics Conference, Dallas, 2012, pp. 141 - 147.

[14] Martin Boldt, Paul Davidsson, Andreas Jacobsson Niklas Lavesson, "Learning to detect spyware using end user license agreements," Knowledge and Information Systems, vol. 26, no. 2, pp. 285-307, January 2010

[15] Fadel Omar Shaban, Spyware Detection Using Data Mining for Windows Portable Executable Files, 2013.

[16] Free Range Security. (2011) [Online]. http://cgi.cs.indiana.edu/~nhusted/dokuwiki/doku.php?id=datasets

[17] Dursun Delen David L. Olson, Advanced Data Mining Techniques.: Springer, 2008.

[18] Michal Kepski Bogdan Kwolek, "Improving Fall Detection by the Use of Depth Sensor and Accelerometer," Neurocomputing, vol. 186, no. C, pp. 637-645, Novmber 2015

[19] Wikipedia. Wikipedia. [Online]. https://en.wikipedia.org/wiki/Precision_and_recall

[20] Tom Fawcett Foster Provost, "Using AUC and Accuracy in Evaluating," in Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, pp. 43-48.

[21] Raffael Vogler. (2015) R news and tutorials. [Online]. http://www.r-bloggers.com/illustrated-guide-to-roc-and-auc/

[22] Iman Lotfi Sara Najari, "Malware Detection Using Data Mining Techniques," Science Publishing Group, vol. 3, no. 6, pp. 33-37, Oct 2014.

[23] APK Downloader. [Online]. http://apps.evozi.com/apk-downloader/

Tawfiq S. Barhoom: is an Associate Professor of Computer Science and head of the computer science department - Faculty of Information Technology at the Islamic University of Gaza, Palestine. Received his Ph.D degree from ShangHai Jiao Tong University (SJTU), in 2004. His current interest research include Secure Software, XMLs security, Web services and its Applications and Information retrieving

Mohammed I. Nasman. Mohammed I.Nasman is professional software developer and trainer.
Mohammed has completed his master thesis from Islamic University of Gaza, his research interests lie in the area of programming languages for topics includes: software engineering, security and mobile platforms.