Proceedings of the International Conference on Intelligent Computing and Communication for Smart World (I2C2SW-2018)
14-15 December 2018

Comparative Analysis of Feature Selection Methods and Machine Learning Algorithms in Permission based Android Malware Detection

M. Nivaashini
Department of CSE
Bannari Amman Institute of Technology
Erode, India
nive19794@gmail.com

R. S. Soundariya
Department of CSE
Bannari Amman Institute of Technology
Erode, India
rssoundariya@gmail.com

H. Vidhya Shri
Department of CSE
Bannari Amman Institute of Technology
Erode, India
vidhyashri92@gmail.com

P. Thangaraj
Department of CSE
Bannari Amman Institute of Technology
Erode, India
thangarajp@bitsathy.ac.in

***Abstract* — The most anticipated cell phone working frameworks available in the market is Google android cell phone board. The open source android board raises trivial problems related to malevolent applications (Apps) and enables designers to take full preferred standpoint of the portable activity framework. On one hand, the distinction of android assimilates consideration of most engineers for building up their applications on this board. Then again, the expanded quantities of utilizations, readies an appropriate inclined for a few clients to create distinctive classes of malware and embed them in Google android advertise or other outsider markets as kindhearted applications. The issue of identifying such malware presents an elite test because of the confined assets accessible and insufficient benefits conceded to the client, yet additionally introduces extraordinary open door in the required metadata connected to every application. Consequently, in this work, android malwares are identified based on the permissions it demands from the client. A few machine learning calculations are being utilized in the discovery of android malware based on the group of permissions empowered for each application. This paper makes an endeavor to examine the execution of different attribute selection methods, like Relief Attribute Evaluator, Gain Ratio Attribute Evaluator, Correlation based Feature Subset Evaluator (CFS), Chi-Square (CH) examination and various machine learning calculations, as Naïve Bayes (NB), J48, Random Forest (RF), Support Vector Machine (SVM), Multi-Layer Perceptron based Neural Network (MLPNN), k-Nearest Neighbor (kNN) and hence an arrangement of results acquired for permission based malware recognition and categorization demonstrates that Chi-Square attribute selection technique and SVM machine learning calculation are overtaking the other feature selection and machine learning methods correspondingly.**

Keywords— Android, Malware Detection, Permission, Feature Selection, Machine Learning.

I. INTRODUCTION

In forthcoming years almost all smartphones comes with the Android operating system and contributes 85% global market share [1][2]. As many users prefers to use Android based handheld device with third party applications security is the major factor to be considered from the perspective of both the user and the service provider. As of late, professionals and specialists have seen the development of an assortment of Android malware. Unlike iOS, android permits many open sources, for example, Google play store, Torrents, Direct downloads, or Third-party markets, and so on [3].

The performance of several feature selection techniques, like, Gain Ratio Attribute Evaluator, Relief Attribute Evaluator, Correlation based Feature Subset Evaluator (CFS), Chi-Square (CH) analysis and machine learning techniques, like like Naïve Bayes (NB), J48, Random Forest (RF), Support Vector Machine (SVM), Multi-Layer Perceptron based Neural Network (MLPNN), k-Nearest Neighbour (kNN) has been compared in this paper[4].

Attackers attempt to draw the clients into running malignant code by making use of the authority for packaging of malware as a simple undertaking task [5]. Noxious payload is utilized in repackaging prevalent applications [6].

Accordingly, malware assaults have reached out to cell phones and it has turned into a need to shield ourselves from such assaults. Android board suppliers has actualized a great deal of safety efforts to stop malware utilizing customary mark based and change-based malware location strategies, which are not ready to adapt to new kinds of malware assaults. With the end goal to conquer this issue, this paper displays a useful and successful peculiarity based malware discovery framework by separating attributes from the permissions of android applications and apply different attribute selection and machine learning procedures for precise recognition and order of both known and obscure android malwares.

The proposed strategy has basically four phases. Initially, from the manifest file of the android applications the permission fields are extricated. Second, an information base of the considerable number of permissions for both ordinary and malware information is built up. Third stage incorporates different attribute selection methods for recognizing effective attributes from the database. Finally to group and distinguish the malware in android applications, the machine learning calculations are utilized [7].

The paper is organized as various sections. The related work IN android malware detection using various attribute

selection and machine learning techniques is described in section II. The overview of the malware analysis and the implementation methodology used in the malware analysis along with the dataset description is given in section III. The results and the findings from analysis using different attribute selection and machine learning techniques are discussed in Section IV. The conclusions and further research scope of malware detection and analysis are presented in section V.

II. RELATED WORK

A. Machine Learning Techniques and Attribute Selection Methods used in android malware detection

Past research used various grouping techniques for ordering malevolent Android applications by performing static examination. Aung et. al, implemented an investigation based on the permissions of the applications by separating the permission set from the significant AndroidManifest.xml documents at Google's Play Market, as expressed in the past segment [8]. With the end goal to limit the attribute size, Information Gain was utilized for attribute determination and classification techniques like J48, Classification and Regression Tree (CART) and Random Forest characterization calculations were implemented to the dataset of reduced size, and measurements like precision, true positive, false positive and recall rates were investigated for the performance analysis. As per the result outcomes, lower false positive rates and higher accuracy rates indicates that CART is over performed by Random Forest and J48.

Peiravian et. al, considered a feature-based learning framework by combining API calls with permissions as features for analyzing the behaviour of the chosen Android applications. The framework includes classification techniques like bagging, SVM and J48. From the results it was inferred that the efficiency of malware identification was improved by combining API calls with the requests for permissions [9].

Glodek et. al, proposed a multi-modular system by including extra properties like native code, intents, repeated groupings of permission requests and receivers of broadcast to the attribute set. The datasets utilized for examination were Android Malware Genome Project which is a public data and an arrangement of generous applications from outsider markets. Random Forest grouping was implemented on the two arrangements of information and the outcomes exposed that the proposed multi-modular system beat existing malware tools, notwithstanding for unique malware programming [10].

Yerima et. al, proposed a static code investigation on Malware Genome Project information arranged Android applications with the assistance of Bayesian classifier. By carrying out reverse engineering for the .apk records, the attributes corresponding to permissions, Linux framework instructions and API calls were separated . To choose attributes from the two classes of utilizations ie benign or malicious, Mutual Information (MI) expansion method was utilized [11]. Outcomes revealed great discovery rates than the well-known mark constructed antivirus programming in light of a similar arrangement of malware tests. Sato et. al, proposed a strategy for recognizing flaws in android malware by breaking down the data that is available just inside manifest documents utilizing J48 Decision Tree calculation [12].

Malware identification incorporate static investigation as well as dynamic examination. Among different techniques, a methodology, called Crowdroid, utilized group sourcing for getting application execution follows. The examination connected a conduct structure to separate Trojan steeds from benevolent applications that had a similar name and a similar form yet a unique powerful conduct. A proficient malware ID procedure was proposed by Burguera et. al, in which the group sourcing application was transferred to Google's Play Market and the calculations use k-means technique for assembling attribute vectors form the framework calls [13].

Shabtai et.al proposed a framework called andromaly based on the host-based behavioural analysis for continuous events and features monitoring from the mobile devices. Feature selection was done using Information gain, Chi-square and Fisher score. After feature selection, classification algorithms like decision tree, histogram, Naïve Bayes, k-means, Bayesian networks and logistic regression are applied [14].

The proposed paper focuses on static examination procedures by broadening the past research work, particularly by presenting a general class of attribute selection techniques and machine learning grouping calculations. Six sorts of characterization calculations, in particular Naïve Bayes (NB), J48, Random Forest (RF), Support Vector Machine (SVM), Multi-Layer Perceptron based Neural Network (MLPNN), k-Nearest Neighbor (kNN) are utilized for an execution examination. Besides, four attribute selection techniques, to be specific Gain Ratio Attribute Evaluator, Relief Attribute Evaluator, Correlation based Feature Subset Evaluator (CFS), Chi-Square (CH) investigation which are subset-based and attribute based, in addition these attribute selection techniques are assessed with their execution commitment to the classification techniques. In the accompanying segment, the informational index has been depicted before presenting the attribute selection techniques and the classification techniques.

III. METHODOLOGY

A. Overview

1)Malware

A program or a product can be stated as malware when it has terrible, unapproved, and unlawful conduct. A portable client can scarcely distinguish the intensions of the application dependent on its authorizations. The clients frequently introduce applications from unapproved sources and welcomes inconvenience. The malware applications normally are customized by the aggressors to take accreditations of money related exchanges, messages, person to person communication data, key stroke data, camera pictures, neighbourhood record framework data and

so on., [15] [17]. Along these lines a portable security application is required that can recognize and distinguish veritable and dangerous applications dependent on permissions that the applications demand.

2)Andriod Malware Analysis

The two techniques which are used to analyse Android malware are static and dynamic.

a)Static Analysis

Static investigation, before execution a program is broken down to check whether it is malignant or not. [17] In static investigation, the discovery board incorporate byte-grouping n-gram, string mark, control stream chart, syntactic library cell and operational code (opcode) recurrence dissemination and so on [8] this examination as a noteworthy disadvantage of dynamic code stacking and code jumbling. [16] In advance undertaking static examination, executable document must be unscrambled and unloaded.

b)Dynamic Analysis

It is utilized to examine the application conduct and noxious code. [15] This is to be performed in a controlled condition by collaborating with a framework. The significant pros of this examination utilized to records the application practices and recognizes dynamic code stacking amid the run-time process. [17] As a result of the overhead of implementing the application, difficult to actualize similarly static examination. This paper manages discovery of android malware utilizing the manifest document permissions that can support in dynamic identification of malware amid establishment.

B. *Dataset*

From the dataset it is reported that out of 3258 android applications only 2999 are the ordinary applications and 1999 are the malware applications. 2015 and 2016 application information from Google play store are utilized for gathering ordinary applications [18] and standard malware informational indexes [19] are utilized in the assessment.

C. *Data Preparation*

1)Attribute Extraction

The apk document must be removed from the .apk expanded record with the end goal to discover the permission information of an application, and afterward the AndroidManifest.xml document must be perused. Consequently, the present work utilizes APKTOOL [20] to remove the apk document into the Android Manifest record and the Smali document. At that point, the permissions characterized in the Android manifest document are analyzed. This can be used to extricate kind of permissions which are required to run the application. Gather the permissions from each application and make an excel sheet. To end with alter it into WEKA.ARFF file format. Fig. 1 delineates the model attribute vector of an ordinary application permission and malware application permission. The unique separated list of attributes contains around 182 attributes acquired from various android mobile applications.

2)Attribute Selection

Attribute Selection is executed with the end goal to limit the dataset dimension by eliminating the attributes (properties) which are not useful to be utilized in the examination. The attributes are picked by their portrayal ability of all the dataset. Effective attribute determination strategies present execution improvements by limiting the dataset dimension and the measure of time engaged with grouping examination. Attribute selection techniques can be sorted into two approachs, namely,attribute-based and subset-based attribute selection strategies .In attribute based attribute selection strategies, the attributes are assessed independently and autonomous of different attributes. It is mostly dependent on allowing for class attributes. Conditions among the attributes are not measured, but rather the connection to the class attribute is considered. Conversely, subset-based attribute selection techniques focus on making arbitrary attribute subsets and the subset that superlatively characterize the entire attributes that has been chosen. Conditions among the attributes are taken into account.

```
0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,normal
0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,1,0,0,1,0,1,0,0,0,0,0,0,0,0,1,0,0,0,1,0,1,0,0,0,
0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,1,0,0,0,1,0,0,0,0,malware
```

Fig. 1. Sample attribute vector of normal and malicious permission

a) Gain Ratio Attribute Evaluator

In this method the significance of an attribute is estimated by calculating the gain proportion concerning the class. It needs a class attribute to quantify attributes. In the projected work, the generous/malware trademark is utilized for the class attribute and furthermore, the quantity of attributes must be determined in this method. Consequently, the quantity of attributes was independently limited by the accompanying plan.

$$\text{Gain Ratio (Class, Feature)} = (H(\text{Class}) - H(\text{Class} \mid \text{Feature})) / H(\text{Feature})$$

a) Relief Attribute Evaluator

In this method the significance of an attribute is evaluated by considering and examining an attribute more than once for an example. It is applicable to both continuous and discrete information classes [21]. The strategy likewise needs a class attribute and a predefined amount of attributes.

b) *Correlation based Feature Subset Evaluator (CFS)*

This strategy estimates the importance of a subgroup of features by making an allowance for the singular prescient capacity of each attribute accompanied by the level of repetition between them. Subgroups of attributes that are profoundly related with the class, though taking less intercorrelation, are favoured [22]. It doesn't need a class attribute since it chooses the subgroup arbitrarily by utilizing a predetermined amount of attributes.

c) Chi-Square (CH) analysis

The X^2 is the feature selection method which is used mostly. In statistics, the independence of two events is tested by X^2 test, and two events A and B are defined to be independent if P(AB) = P(A)P(B) or it can be expressed as P(A|B) = P(A) and P(B|A) = P(B) . The occurrence of the term and occurrence of the class are the two events in feature selection.

The four attribute selection strategies portrayed in this section are independently kept running on the dataset. It consists of 3258 android applications with various attributes.

D. Detection and Classification

With the end goal of malware recognition, the accompanying machine learning categorization techniques are kept running on the dataset with decreased attributes set, and utilize Weka [23] to implement the machine learning techniques.

1)Support Vector Machine (SVM)

The SVM partition the n-dimensional space of the information into two areas by utilizing a hyperplane and it expands the edge between the two locales isolating two classes of tests. The separation between the instances of the two classes is characterised by the edge and its calculation dependents on the separation between the nearest occurrences of the two classes, which are called support vectors [24].

2)Random Forest (RF)

The knowledge behindhand this classification technique is to make a forest of random trees [25].

3)k-Nearest Neighbour (kNN)

This classifier delays the classification until a query is made before that it tries to generalize the dataset [26] so it is called as lazy classifier, The classifier which performs the classification based on comparison methods is KNN classifier. The training and test instance are compared to identify the class of the nearest k instances .

4)Naïve Bayes (NB)

The Bayes classifier uses the estimator classes for calculations Based on the examination training data, numeric estimator accuracy is chosen [27].

5)Multi-Layer Perceptron based Neural Network (MLPNN)

MLPNNs are very quick, simple to utilize, work and actualize, and need a little dataset therefore making them genuinely famous to utilize in data examination [28]. The system is signified by hubs, which have two yields and sources of info vectors, furthermore the hubs are associated together by weights. For preparing the data collection, MLPNNs utilizes back-proliferation.

This calculation creates either a pruned or an unpruned C4.5 choice tree [29].

IV. Experimental Results

The assessment of the grouping calculation executions is achieved as far as the accompanying assessment procedures: Accuracy (ACC), True Positive Rate (TPR), False Positive Rate (FPR), and Precision. These methods are gotten from four fundamental estimates that are depicted beneath.

True Positive (TP): It indicates the no. of accurately classified genuine android applications.

True Negative (TN): It indicates the no. of accurately classified malicious android applications.

False Positive (FP): It indicates the no. of inaccurately classified malicious android applications.

False Negative (FN): It indicates the no. of inaccurately classified genuine android applications.

The resulting assessment procedures are defined in terms of the simple procedures, as offered beneath:

Accuracy: The proportion of accurately classified applications. In other words, it shows the proportion of accurately classified samples.

Accuracy (ACC) = (TP + TN) / (TP + TN + FP + FN)

TPR: The proportion of accurately classified genuine android applications. TPR is also called Recall.

True Positive Rate (TPR) = TP / (TP + FN)

FPR: The proportion of inaccurately classified malicious android applications.

False Positive Rate (FPR) = FP / (TN + FP)

Precision: It is the proportion of recovered samples that are significant. It is also called positive predictive value.

Precision = TP / (TP + FP)

The estimation of the selected attribute selection techniques is depicted in the following section.

Table 1 shows the decrease of attribute set dimension by four various attribute selection methods of which Chi Square analysis produce 34 best significant attributes. The reduced attributes can be convoluted in the malware discovery and classification process.

TABLE 1. REDUCED FEATURE SET BY VARIOUS FEATURE SELECTION METHODS

Feature Selection Techniques	No. of feature selected
Gain Ratio Attribute Evaluator	41
Relief Attribute Evaluator	48
Correlation based Feature Subset Evaluator (CFS)	43
Chi-Square (CH)	**34**

Table II and Fig. 2 depicts the experimental outcomes of all the six classifiers for various set of attribute sets nominated by various attribute selection techniques. The table shows that SVM have achieved better performance over other classification algorithms with the decreased attribute set acquired from the Chi Square analysis attributes selection technique. The regular accuracy was detailed as

99.9% that is the maximum than other classifiers and also maximum than the existing work revealed in [30] [31]. In terms of computational complexity, better performance is obtained in Naïve Bayes classifier. In android malware identification the performance of Multilayer perceptron (MLP) neural network classifier is closer to the SVM and the computational complexity of MLP classifier is also reduced.

TABLE II. PERFORMANCE COMPARISON OF STATIC MALWARE DETECTION WITH VARIOUS MACHINE LEARNING CLASSIFIERS

Machine Learning Models	ACC (%)	Precision (%)	TPR (%)	Computation Time (sec)
NB	99.1	99.1	99.1	**0.065**
J48	98.7	98.7	98.7	0.16
RF	99.5	99.5	99.5	1.45
SVM	**99.9**	**99.9**	**99.9**	0.15
MLPNN	99.6	99.6	99.6	356.8
kNN	97	97	97	0.36

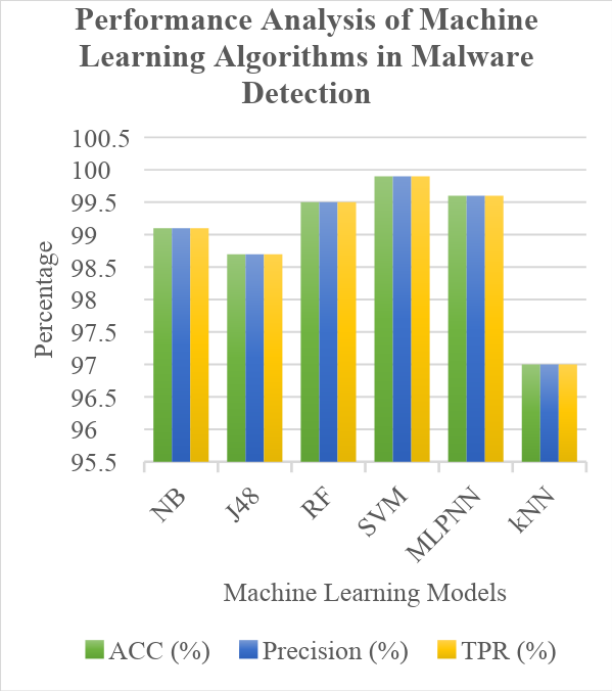

Fig. 2. Performance comparison of Static Malware Detection with various Machine Learning Classifiers

V. CONCLUSION & FUTURE WORK

In this paper for the process of android malware identification the performance of different feature selection techniques and machine learning algorithms are examined. For classifying and checking whether an android applications is malware or not, the proposed system make use of various machine learning techniques .To access the performance of the proposed model, various permissions and features from numerous downloaded mobile applications from the android market have to be extracted. From the experimental analysis, it was found that SVM has performed well over other methodologies when combined with the Chi square analysis feature selection technique, regarding classification accuracy. When computational complexity is compared, Naïve Bayes classifier showed improved results in categorizing malware data set. The results of the proposed work is also compared with existing android malware identification and the results show the greater classification accuracy. Hence, the future research should concentrate on the static analysis of android malware by using other static attributes that are available in an apk files, by also including API calls with the permissions as a requirement of higher resource in the process of analysis. Then, the future research may focus on identifying emergent clusters in the dataset using on clustering analysis. Finally, deep learning feature reduction and classification algorithms can be used in the dynamic analysis of malware.

REFERENCES

[1] Shiladitya Ray, "Android phones will have 85% global market share in 2018 ", August 31 2018, Available: https://www.newsbytesapp.com/timeline/Business/30835/137527/the-global-smartphone-market-in-2018.
[2] Global mobile OS market share in sales to end users from 1st quarter 2009 to 2nd quarter 2018, Available: https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/
[3] Madihah Mohd Saudi and Zul Hilmi Abdullah, "An Efficient Framework to Build Up Malware Dataset," International Journal of Computer, Electrical, Automation, Control and Information Engineering, vol. 7, pp. 1104-1109, 0ctober 7 2013.
[4] S. P. Choudhary and Deepti Vidyarthi, "A Simple Method for Detection of Metamorphic Malware using," ScienceDirect, pp. 265-270, 2015.
[5] Abhay pratap singh and S.S handa, "Malware detection using data mining techniques," International Journal of Advanced Research in Computer and Communication Engineering, vol. 4, no. 5, pp. 375-380, May 2015.
[6] Xialoei wang, yuexiiang, and zeng yinghi, "Accurate malware detection and classification in the cloud," Spinger plus, pp. 1- 23, 2015.
[7] Babu Rajesh V, Phaninder Reddy, Himanshu P, and Mahesh U Patil, "Androinspector: A System for comprehensive analysis of android applications," International Journal of Network Security & Its Applications (IJNSA), vol. 7, pp. 1-21, September 2015.
[8] Aung, Z., & Zaw, W., "Permission-based Android malware detection.", International Journal of Scientific & Technology Research, vol. 2, no. 3, pp. 228-234, Mrach 2013.
[9] Peiravian, N., & Zhu, X., "Machine Learning for Android Malware Detection Using Permission and API Calls", IEEE 25th International Conference on Tools with Artificial Intelligence (ICTAI), pp. 300-305, November 2013.
[10] Glodek, W., & Harang, R., "Rapid Permissions-Based Detection and Analysis of Mobile Malware Using Random Decision Forests", IEEE Military Communications Conference, pp. 980-985, November 2013.
[11] Yerima, S. Y., Sezer, S., McWilliams, G., & Muttik, I., "A new android malware detection approach using Bayesian classification", IEEE 27th International Conference on Advanced Information Networking and Applications (AINA), pp. 121-128, March 2013.
[12] R. Sato, D. Chiba and S. Goto, "Detecting Android Malware by Analysing Manifest Files", Proceedings of the Asia-Pacific Advanced Network, Vol. 36, p. 23-31., 2013.
[13] Burguera, I., Zurutuza, U., & Nadjm-Tehrani, S. "Crowdroid: behavior-based malware detection system for android", Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices, pp. 15-26, October 2011.
[14] Shabtai, A., Kanonov, U., Elovici, Y., Glezer, C., & Weiss, Y., ""Andromaly": a behavioural malware detection framework for

android devices", Journal of Intelligent Information Systems, vol. 38, no. 1, pp. 161-190, 2012.

[15] Mayuri Magdum and Sharmila.K.Wagh, "Permission Based Android Malware Detection System using Machine Learning Approach," International Journal of Computer Science and Information Security (IJCSIS), vol. 14, no. 2016.

[16] Prajakta D. Sawle and Prof. A. B. Gadicha, "Analysis of Malware Detection Techniques in Android," International Journal of Computer Science and Mobile Computing, vol. 3, pp. 176-182, March 2014.

[17] Imtithal A. Saeed, Ali Selamat, and Ali M. A. Abuagoub, "A Survey on Malware and Malware Detection Systems," International Journal of Computer Applications, vol. 67, April 2013.

[18] Ridhima Seth and Rishabh Kaushal, "Dissecting Android Malware: Characterization and Evolution," IEEE, 2012.

[19] "android-apktool, A tool for reverse engineering Android apk files" URL http://code.google.com/p/android-apktool

[20] Kononenko, I., "Estimating attributes: analysis and extensions of RELIEF", Machine Learning: ECML, Springer Berlin Heidelberg, pp. 171-182., 1994.

[21] Hall, M. A., "Correlation-based feature selection for machine learning", Doctoral dissertation, The University of Waikato, 1994.

[22] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," SIGKDD Explorations, vol. 11, pp. 10–18, 2009.

[23] V. Vapnik, The nature of statistical learning theory. Springer, 2000.

[24] Breiman, L., "Random Forests", Machine Learning. vol. 45 no. 1 pp.5-32, 2001.

[25] S. Vijayarani and M. Muthulakshmi, "Comparative analysis of bayes and lazy classification algorithms," International Journal of Advanced Research in Computer and Communication Engineering, vol. 2, pp. 3118–3124, 2013.

[26] John, G. H., & Langley, P. "Estimating continuous distributions in Bayesian classifiers", In Proceedings of the Eleventh conference on Uncertainty in artificial intelligence, pp. 338-345,1995.

[27] U. Orhan, M. Hekim, and M. Ozer, "EEG signals classification using the K-means clustering and a multilayer perceptron neural network model," International Journal of Advanced Research in Computer and Communication Engineering, vol. 38, pp. 13 475–13 481, 2011.

[28] Quinlan, J.R., "C4.5: Programs for Machine Learning", Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[29] Ravi Kiran Varma P, Kotari Prudvi Raj, K. V. Subba Raju, "Android mobile security by detecting and classification of malware based on permissions using machine learning algorithms", IEEE International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), 2017.

[30] Ugur Pehlivan, Nuray Baltaci, Cengiz Acartürk, Nazife Baykal, "The Analysis of Feature Selection Methods and Classification Algorithms in Permission Based Android Malware Detection", IEEE Symposium on Computational Intelligence in Cyber Security, 2014.