Mobile Malware Detection using Anomaly Based Machine Learning Classifier Techniques

Saja Ibraheem Hani* **and Naji Matter Sahib**

Computer Science Department – College of Science – University of Diyala

sajaibraheemhani@gmail.com

Received: 20 July 2020 **Accepted: 21 December 2020**

DOI: https://dx.doi.org/10.24237/djps.17.02.544B

Abstract

Smartphones have become essential in our daily lives. Many works can be done by using it like, browse the internet, and download many applications for each device through the available store. As a result, the number of malware applications downloaded also increases. These malware carries out various activities behind the scenes, such as breach of confidentiality, breach of privacy, loss of confidentiality, system breakdown, theft of sensitive information, etc. Many types of research and studies have proposed different techniques to detect malicious programs, but these measures contain weak points, which are illustrated by efficiency, speed, and lack of comprehensiveness. In this paper, a proposed system is designed and implemented to detect malware in smartphones using anomaly detection technology that begins to extract the important features that play an effective role in detecting malicious code and applying machine learning algorithms. The proposed system has been tested using a hybrid Genetic algorithm, and the Support Vector Machine data has been registered with an accuracy of (0.9282%). The experimental results indicate that the proposed system has a high average accuracy rate compared with other existing methods where there is a (0.8848%) average accuracy using Probabilistic Neural Network, while the average accuracies of (0.8835%) and (0.8715%) respectively with Support Vector Machine and K-Nearest Neighbors.

Keywords: malware, anomaly detection, learning machine technique, hybrid algorithm.

Mobile Malware Detection using Anomaly Based Machine Learning Classifier Techniques

Saja Ibraheem Hani and Naji Matter sahib

كشف البرامج الضارة للهاتف الجوال باستخدام الشذوذ القائم على تقنيات مصنف على التعليم الآلي

سجى ابراهيم هاني و ناجي مطر سحيب

قسم علوم الحاسبات – كلية العلوم – جامعة ديالى

الخلاصة

أصبحت الهواتف الذكية ضرورية في حياتنا اليومية. يمكن القيام بالعديد من الاعمال بأستخدامه مثل تصفح الإنترنت وتنزيل العديد من التطبيقات لكل جهاز من خلال المتجر المتاح. نتيجة لذلك، يزداد أيضًا عدد تطبيقات البرامج الضارة التي يتم تنزيلها. تنفذ هذه البرامج الضارة أنشطة مختلفة وراء الكواليس؛ مثل السرية وانتهاك الخصوصية وفقدان السرية وانهيار النظام وسرقة المعلومات الحساسة وما إلى ذلك. اقترحت العديد من البحوث والدراسات تقنيات مختلفة للكشف عن البرامج الخبيثة، لكنها احتوت على نقاط ضعف، والتي تتضح من خلال الكفاءة والسرعة وعدم الشمولية. في هذا البحث، تم تصميم وتنفيذ نظام مقترح للكشف عن البرامج الضارة في الهواتف الذكية واستخدام تقنية الكشف عن الشذوذ، التي تبدأ في استخراج الميزات المهمة التي تلعب دورًا فعالًا في اكتشاف التعليمات البرمجية الضارة وتطبيق خوارزميات التعليم الآلي. تم اختبار النظام المقترح بأستخدام خوارزمية جينية هجينة مع خوارزمية الة المتجهات الداعمة، وتم تسجيل اعلى دقة (0.9282%). أشارت النتائج التجريبية إلى أن النظام المقترح يتمتع بمتوسط معدل دقة مرتفع مقارنة بالطرق الأخرى الموجودة حيث بلغ متوسط الدقة (0.8848%) باستخدام الشبكة العصبية الاحتمالية، بينما كان متوسط الدقة (0.8835) و (0.8715) باستخدام دعم الة المتجهات و K -أقرب الجيران على التوالي.

الكلمات المفتاحية: البرمجيات الخبيثة، اكتشاف الشذوذ، تقنيات آلة التعليم، خوارزمية الهجينة.

Introduction

Mobile devices have become an open platform to implement various applications. Mobile app downloads worldwide are expected to reach 224,801 billion in 2016 and will increase continuously [1]. Provides the rapid growth of the smartphone industry and the rapid promotion of mobile communication technologies, more and more consumers use smartphones to access the Internet and consume various services. Mobile apps provide a great convenience for our daily life by providing instant access to a wealth of information via the Internet, support for ongoing communications anywhere, and provide various functionalities. The rapid growth of mobile apps plays a critical role in the future success of mobile Internet, and the economy. Smartphones usually store user's private data such as photos, messages, and personal credentials. Get, they become the target of many malicious attackers [2, 3]. Furthermore, to adapt the unknown sessions and use the rule-based in integration with the attacks that are a very important aspect is the identification of the classification algorithm of the most reliable detection accuracy. Therefore, the strongest classifiers are identified through the evaluation of the activeness and detection accuracy of all machine-learning algorithms. Modifying the default input values can enhance the efficiency and accuracy of a classifier. However, enabling an equivalent comparison between the classifiers dictated the implementation of the classifiers with their default input values. Many studies and researches have emerged to discover and treat malicious programs based on the artificial intelligence algorithm. Such as K-Nearest Neighbors (k-NN), naive Bayes, Random Forests (RF), Support vector machines (SVM), and Genetic algorithm [4].

Related works

A lot of works and studies have been proposed for malware detection and analysis. Nath et al., It is difficult to recognize targeted malware by antivirus, IDS, IPS, and custom malware detection tools. Attackers leverage compelling social engineering techniques along with one or more zero-day vulnerabilities. Our propose a comparison between different machine-learning techniques to detect such dreadful malware [5]. Baltacı, et al., The main purpose of the study is to investigate the contribution of other application market metadata to the detection of malicious applications in addition to requested permissions.

Hence, the information of applications presented on the official market when a user wants to download them was used as the feature set for training supervised classification algorithms. This drawback of the proposed model may be caused by the method used to label applications as malicious or benign. In their study, both the malicious and benign applications are labeled by querying the analysis results of them on VirusTotal, [6]. Moutaz Alazab et al., the study proposed a system that classifies mobile marketplaces applications with the use of real-world datasets, which performs an analysis for the source code to identify malicious applications. Their study proposed a system based on feature selection and supervised machine-learning algorithms for detecting mobile malware in the marketplace [7].

Anomaly detection

The current study aims to find whether or not Google Play, Google's, and Android's official application market, metadata of Android applications assist in explaining the malicious behaviors when joined with user's permissions analysis. Therefore, it was required to collect the metadata of the application on Google Play. First, a kind of web automation and testing tool and a browser-based macro recorder, iMacros was used to collect Google metadata and class (or target) values of Android applications. The processes of data collection were recorded, and their macrocodes obtained by iMacro were embedded into Microsoft Visual Basic for repetitive data collection tasks. However, fetching an enormous number of application data by clicking one by one consumes both time and effort significantly. Hence, a more practical solution was performed, and that is utilizing a web crawler and querying data from the servers of Google Play directly. The use of Google Play Crawler was to acquire the permissions requested by applications and to download them as .apk file. Permissions collected for the top unpaid applications, there are 851 different permissions as a whole in each application category, and some of them consist of developer-defined permissions. When users visit the Android application's page on Google to download, this crawler gives them incomplete information. Therefore, separately, a Java application was applied for collecting other metadata of applications. The Applications were downloaded and their Google Play information was gathered on 17 June 2014. The consumed time for data collection is 12 hours approximately. Form each application category, the dataset includes top free ones, and they make 17244 applications on the date of collection. The following information is included in the mentioned Google metadata of applications in this study: Application category, Developer's name, Developer type, rating stars, Average rating of application, Publish date of an application, Application size, Minimum required Android OS version, Application content rating, Number of application downloads in form of range. After market-related information of applications is collected (including permissions), the second step is acquiring the target values of applications (malicious or benign) to have the ability to train a supervised classification algorithm.

Machine learning technique for classifying dataset

In this study aimed to find an estimated performance of the detection model, which is proposed, in the current study. The use of a genetic algorithm and SVM classifier came for a faster result generation and comparison. The first step included collecting the entire dataset of applications from the official market; it was used for applying the Genetic algorithm and SVM learning algorithm. Data of relation to application market was gathered through writing a Java application for all application classes defined by Google Play and for top free applications under each of them.

A Google Play crawler was used to collect the permissions requested by the application at the time of installation, it also is used to download applications again. Downloading the applications is carried out to figure their hash values to query them via an online free Antivirus (AV) engine (Virus Total). An application is defined as malicious if 2 or more AV engines recognize it as malicious, otherwise, it is benign for the initial phase.

The investigation of the construction of official market metadata was performed using various datasets; one of which includes market metadata and users' permissions, while the other contains only the users' permissions. The same number of instances is included in these two datasets, consisting of 4512 malicious and 12719 benign applications. Show in figure (1) anomaly classifying.

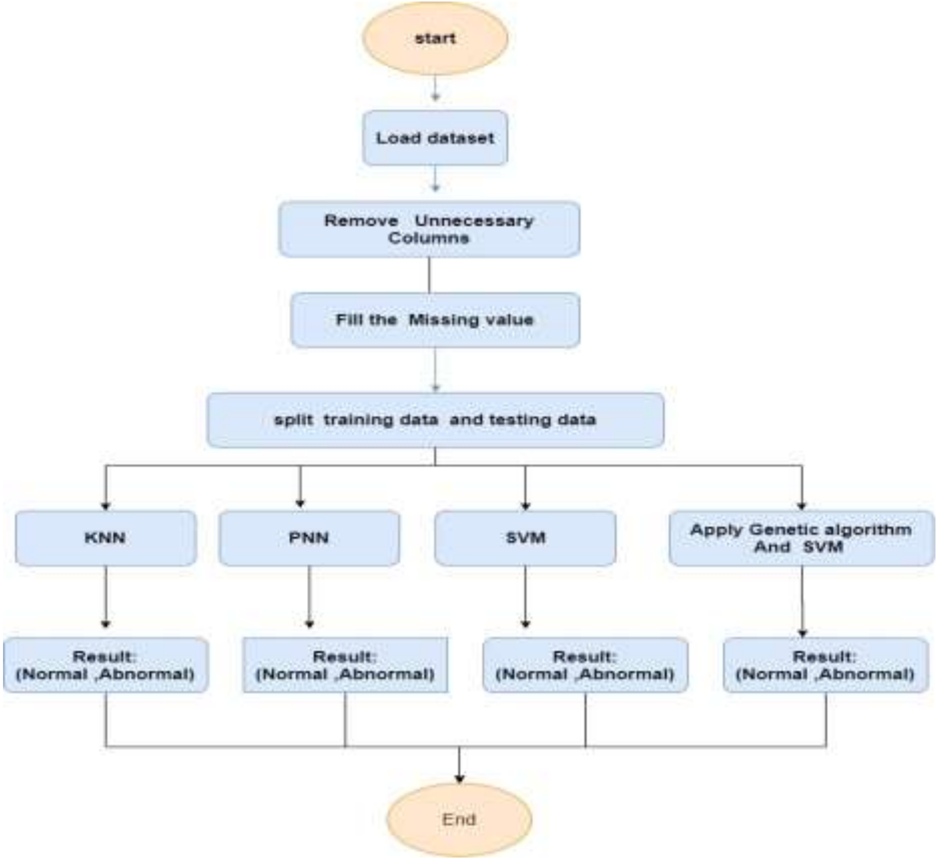
start
Load dataset
Remove Unnecessary Columns
Fill the Missing value
split training data and testing data
KNN
PNN
SVM
Apply Genetic algorithm And SVM
Result: (Normal ,Abnormal)
Result: (Normal ,Abnormal)
Result: (Normal ,Abnormal)
Result: (Normal ,Abnormal)
End

Figure 1: General Block Diagram (anomaly classifying)

Classification Techniques

Malware detection can be seen as a problem of classification unknown malware types should be clusterized into several clusters, based on certain properties, identified by the algorithm. On the other hand, having trained a model on the wide dataset of malicious and benign files, we can reduce this problem to classification. For known malware families, this problem can be narrowed down to classification only – having a limited set of classes, to one of which malware sample belongs, it is easier to identify the proper class, and the result would be more accurate.

a) SVM Technique

In the proposed work, apply SVM to detection Malware in Android, Trained Classification SVM classifiers store prior probabilities, parameter values, support vectors, algorithmic implementation information, and training data. These classifiers are used for tasks like fitting a score-to-posterior-probability transformation function. In SVM to get optimize training performance these given various parameters.

Training Data Simulation 70% of the database was used for training. The benign class contained 420 samples, while the malignant class included 180 samples of the training data. Table (1) presents the classification results of the training set by SVM.

Table (1): Classification of test SVM

	Normal	Abnormal
True	442.0000	87.0000
False	0	70.0000

performances of the SVM classifiers obtained with linear functions with and without feature selection in terms of classification the evolution of SVM (True positive= 414, True Negative= 108, False positive = 35, False Negative = 42, Accuracy = 87.15%, TPR= 90.79%, FPR= 24.48%, TNR= 75.52%, FNR = 9.21%, Precision =92.20%) . Show the figure (2)

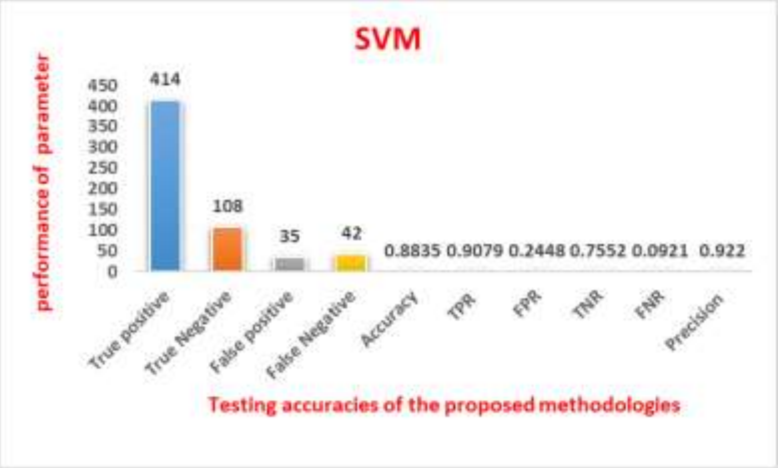

Figure 2: Training and test accuracy using SVM

b) Genetic Algorithm with SVM Technique

The learning model is built to categorize the data after preparing the input dataset where the missing values have been filled and cleaned data. In this work, the genetic algorithm used to reduce the data entered into the SVM model, to obtain the highest accuracy and less time in determining the data class. The conditions that must be met are in the chromosome. The stays should not be repeated. The fitness function depends on the SVM function as shown in (1).

Algorithm 1: Genetic Algorithm and SVM Technique

Input: Dataset

Output classification dataset

Step 1: After cleaning and missing value The Android malware detection data consists of (600 raw and 167 column)

Step 2: Random population generation, where 1000 individuals (chromosome) are generated and each chromosome contains 20 cells, as a cell contains the column heading in the used dataset, with no similar numbers being repeated in one chromosome

Step 3: Fitness function, the calculation of the efficiency of each chromosome using SVM (each chromosome contains 20 columns of data is taken ignore the rest and the accuracy of the data entered into the SVM is calculated, knowing that the separation of training data and test data is 70% randomly)

Step 4: Selection operation, two chromosomes are randomly selected as parents.

Step 5: Crossover operation, in this step the two-chromosome exchange is separated and the second part of the first chromosome is replaced by the second part of the second chromosome using (single point) and the formation of sons

Step 6: Mutation operation: from the previous step, two or more numbers may be repeated. In this step, the number repeated is replaced with a new number.

Step 7: Fitness The fitness calculation for children is produced using SVM

Step 8: Update the population

Step 9: Repeat previous operations (from the selection step, reach the desired goal or 100 iterations).

Table 2: Classification of test hybrid Genetic algorithm and SVM

Classification of test hybrid Genetic algorithm and SVM		
	Normal	Abnormal
True	444	112
False	0	43

After achieving feature selection with the use of the genetic algorithm, the numbers of features selected from the dataset with and without feature selection in terms of classification the evolution of hybrid Genetic algorithm and SVM (True positive=444, True Negative=112, False positive= 0, False Negative= 43, Accuracy= 92.82%, TPR= 91.17%, FPR= 0%, TNR=1%, FNR=08.83%, Precision=1%).

c) PNN Technique

The Probabilistic neural network (PNN) is a promising machine learning technique that can be used to forecast financial markets with higher accuracy [10].

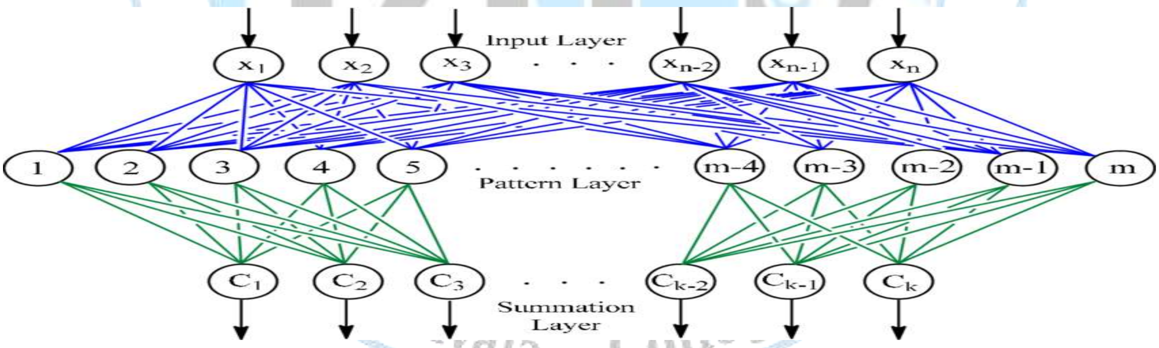

Figure 4: Probabilistic neural network (PNN) general architecture [10]

Training Data Simulation 70% of the database was used for training. 420 samples of the training data belong to the benign class and 180 samples belong to a malignant class. The classification results of the training set by PNN were given in table 3.

Table 3: Classification of training PNN

	Normal	Abnormal
True	443	87
False	0	69

PNN classifier has a nearly perfect accuracy value of 88.86 % when original features without select the best feature or reduce the amount of feature the performance parameter of PNN is (True positive=443, True Negative=87, False positive=0, False Negative=69, Accuracy=88.48%, TPR=86.52%, FPR=0%, TNR=1%, FNR=13.48%, Precision = 1%)

d) KNN Technique

The k-NN classifier is one of the simplest and most widely used in such classification algorithms. The relative simplicity of the *k*-NN search technology makes it easy to compare the results from other classification techniques to *K*-NN results. Training Data Simulation 70% of the database was used for training. 420 samples of the training data contained by benign class while 180 samples of the training data contained by malignant class. The classification results of the training set by K-NN were given in table 4.

Table 4: Classification of test KNN

	normal	Abnormal
True	414	108
False	35	42

K-NN classifier has a nearly perfect accuracy value of 88 % when original features without select the best feature or reduce the amount of feature the performance parameter of K-NN is (True : positive=414, True: Negative=108, False: positive=35, False : Negative=42, Accuracy=87.15%, TPR=90.79%, FPR=24.48%, TNR=75.52%, FNR=9.21%, Precision = 1%).

The performance evaluations of each type of distance and classification rules are chosen function of classification accuracy rate and time classification for each value of the nearest neighbor’s parameter. To validate the results, tests have been carried out. The nearest rule was used for the classification rules to classify a new element. The high rate of classification accuracy was 88% recorded by the algorithm that utilizes Euclidean distance with a value of k = 1. When K increases, the rate of classification accuracy decreases then takes a stable state at approximately 50, with almost 88% classification accuracy rate. However, the optimum result is generated with Euclidean distance (88%), this corroborates with what was presented in the literature. The smallest rate of classification achieved in attrition 50 with 73%. In contrast, Euclidean is a time-consuming classification, these results are illustrated in Figure (7)

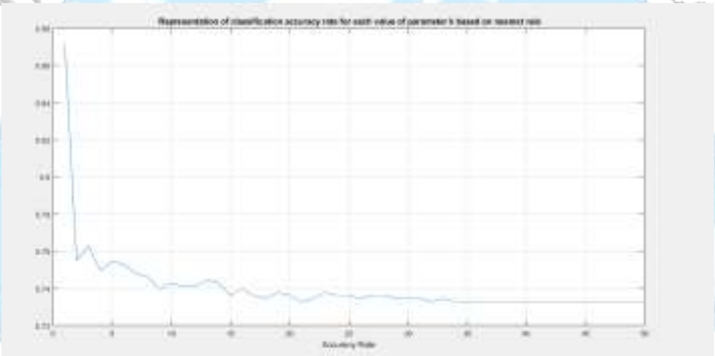

Figure 7: Representation of classification accuracy rate for each value of parameter k based on random rule

Experimental Results

This paper aims mainly to provide a suitable answer to the question: does whether Google Play market metadata plays a crucial role while detecting mobile malware and contributes to the detection model with only Android permissions? For answering this, two baseline datasets with corresponding classification algorithms are compared. Any feature selection algorithm does not accompany the classification algorithms, thus, when results are examined for them, they show that an improvement is gained in half of the prediction accuracy of models due to adding the official market metadata to the model. Addition of market metadata of accuracy 88.65% for the PNN, 87.15% for SVM, 88.35% for K-NN. Table (5) and figure (8) are illustrated the experimental results

Table 5: The Result of Performance Evaluation of machine learning technique

Statistic	PNN	KNN	SVM	Hybrid Genetic algorithm and SVM
True positive	443.0000	414.0000	414.0000	444.0000
True Negative	87.0000	108.0000	108.0000	112.0000
False positive	0	35.0000	35.0000	0
False Negative	69.0000	42.0000	42.0000	43.0000
Accuracy	0.8848	0.8715	0.8835	0.9282
TPR	0.8652	0.9079	0.9079	0.9117
FPR	0	0.2448	0.2448	0
TNR	1.0000	0.7552	0.7552	1.000
FNR	0.1348	0.0921	0.0921	0.0883
Precision	1.0000	0.9220	0.9220	1.0000

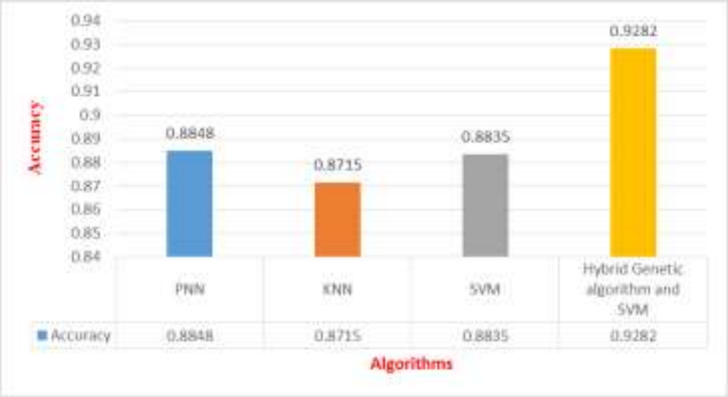

Figure 8: The compare of Algorithms

Conclusions

Many conclusions have been deduced from the obtained test results. Use a hybrid approach solution full protection strategy to increase system confidence and security due to anomaly detection schemes and provide full protection mechanisms in the Android platform. The conventional Machine learning-based network intrusion detection system (NIDS) presented so far is especially realistic in regards to the unstable detection accuracy and variance of detection speed, which are closely linked to the technique of retrieving the critical features of each attack.

Acknowledgments

I am pleased to thank and show my gratitude to those who have helped me to achieve my paper whether through assisting to provide resources or with guidance and presenting my advice.

References

1. C. L. P. M. Hein, K. M. Myo, International Journal of Computer Applications, 181 (19), 29-39 (2018.)
2. Y. Zhou, X. Jiang, IEEE Symposium in Security and Privacy (IEEE S&P), 95-109 (2012)
3. A. M. Memon, A. Anwar, IEEE Security & Privacy, 13(6), 77-81 (2015).
4. A. Moser, C. Kruegel, E. Kirda, Limits of static analysis for malware detection, in: Twenty-Third Annual Computer Security Applications Conference, Dec 10, 2007, 421-430
5. J. Devesa, F. Brezo, J. Nieves, Bringas, A Static- Dynamic Approach for Machine Learning-Based Malware Detection, in Advances in Intelligent Systems and Computing, volume 189, (AISC, 2013)
6. K. Kemalis, T. Tzouramanis, SQL-IDS: a specification-based approach for SQL-injection detection, In: Proceedings of the 2008 ACM symposium on Applied computing, 2153–2158.
7. H. V. Nath, B. M. Mehtre Static Malware Analysis Using Machine Learning Methods, In: International Conference on Security in Computer Networks and Distributed Systems SNDS, 2014, Springer, Berlin, Heidelberg, 440-450
8. N. Baltaci, A Comparison of classification Algorithm for mobile malware detection: market metadata as input Source, M.Sc. Thesis, The Middle East Technical University, 2014.
9. M. Alazab, Electronics, 9(3),435(2020).
10. H. Kurniawan, Y. Rosmansyah, B. Dabarsyah. Android anomaly detection system using machine learning classification, In: Electrical Engineering and Informatics (ICEEI), International Conference, Aug 2015, 288–293.