ISSN 2278-3091

Volume 9, No.1.5, 2020

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse4691.52020.pdf

https://doi.org/10.30534/ijatcse/2020/4691.52020

Android Malware Detection System using Machine Learning

Nuren Natasha Maulat Nasri[1], Mohd Faizal Ab Razak[1*], RD Rohmat Saedudin[2], Salwana Mohamad@Asmara[1], Ahmad Firdaus[1]

[1]Faculty of Computing, Universiti Malaysia Pahang, Malaysia, faizalrazak@ump.edu.my

[2]School of Industrial Engineering,Telkom University,40257 Bandung West Java, Indonesia, roja2128@gmail.com

ABSTRACT

During the past year until now, the amount of malware targeting Android operating system has been rising dramatically. Therefore, Android malware detection are required to detect the malware before getting more serious. The static analysis examines the full code of application meticulously while dynamic analysis identifies the malware applications by monitoring it behaviors. This study proposed a malware detection system by using machine learning approach and aims to detect malware that has attacked Android operating system. In this research paper, the Android malware detection system are trained using five types of classifiers meanwhile WEKA is used for simulation process. The dataset used contains 10k of malware and 10k of benign. The outcomes presented Random Forest classifiers attained highest accuracy result, 89.36% compared to Naïve Bayes which 89.2%. TPR is viewed as detection rate which precisely predicted malware process while FPR is choosed as detection rate which inaccurately predicted normal as malware. To evaluate detection exactness which is good or bad, the area under the curve (AUC) have been applied through this study. The results show that Naïve Bayes has the lowest model complexity as it uses minimal time to build the model. Hence, it can be concluded that achieving reasonable accuracy and effectiveness in classifying unknown malware helps to determine the performance of the classifiers.

Key words: Android, Intrusion, Machine Learning, Malware.

1. INTRODUCTION

Here Nowadays, the usage of mobile devices or smartphones has increasing in our daily and almost all of the people around world own a smartphone. According to Global market share, during second quarter of 2018, there was 88% smartphones in the market have been sold towards end users and that is Android systems [13].

Besides, it is becoming more and more popular because of its portability and convenient to use. For an example, the smartphone contains various types of functions and services like it can hold the personal information and access files that usually been stored in the cloud such as bank account information, email details, password and it also allows the user to interact with each other by sending a message or call. However, with the growth of the Android mobile popularity has brought many security concerns and threats from the attacker that might spread the malware that makes the system act differently than it is supposed to behave. The malware usually sent such fraudulent message and charge the user for their fake services.

According to the Security Threat Report released by Symantec in 2018 [12], the overall target activities that attacked is up by 10 percent in 2017. In fact, by March 12, 2018, there are 4, 964, 460 devices infected by RottenSys malware [4]. This situation desperately needs to find a potential method to detect malware before it harmed more Android smartphones. In this era globalization, people commonly used smartphones in such many ways like using a network connection to interact with the world. For example, online shopping, online banking, and cloud storage. Naturally, there are also has disadvantages by using this kind of network connections towards the user. Like example, the storing of confidential information in smartphones might attract the attacker to use dirty things in order to get user details like spreading malware towards some software or applications that might be installed in their smartphones either they realized or not especially for Android users.

Besides, there are many kinds of existing research that had been proposed to detect the malware by using various types of techniques and methods that implement into the application. For example, Google published an automated scan system for potential malware which is called as Bouncer application [8]. However, there is still has room for improvement of Android malware detection. The reason is the different type of method and techniques will come out with a different rate of error results.

Furthermore, there is still some false alarm occurred on Android devices that tricked the user. For example, there are 600,000 of Android user that have been downloaded the fake guide applications such as Pokemon Go and FIFA mobile. This is because, they are mistakenly downloaded the malware application when they want to seek the guide for the games [14]. These prove that not all the techniques have been successfully developed in order to give protection for Android smartphones. Hence, the lead contributions of this research as follow:

i. To review current issue related to the Android malware detection system.

ii. The evaluation study applied machine learning approach has improved the malware detection system.

iii. To evaluate propose of the system in terms of accuracy of malware detection.

The rest of this paper are sorted as follows. The Section 2 discuss about related work that have been used by previous researcher. Meanwhile, Section 3 explained the details about the research method during experiment and Section 4 evaluates the successfulness in detecting the malware. Lastly, section 5 is the conclusion and future work of this paper.

2. RELATED WORK

Machine learning mainly known as an artificial intelligence (AI) application that provide the system potentiality to automatically learn and upgrade experience without being program explicitly. Therefore, the machine learning approach can provide a solution to improve the decision-making process [9]. Recently, machine learning approach have been used to perform the decision-making task such as text-based on sentiment analysis and pattern recognition, detecting the malware, network intrusion detector and etc. [3][25][26]. There are some machine learning methods as stated in [21].

- Supervised machine learning algorithms able to predict the upcoming events by soliciting the things from the previous learning to the new data using labeled examples. By using this learning, inferred function can be produced to make prediction about the output values which is training dataset after analysis. Besides, this algorithm also able to compare the output with the exact one, intended output and find error to adjust the model accordingly.
- Meanwhile for unsupervised machine learning algorithms, it trained the information neither classified nor labeled. This algorithm study how systems can infer the function to relate with the hidden structure from unlabeled data. Moreover, the system explores the data and draw the inferences from datasets to express the unlabeled data hidden structures without figure out the exact output.
- Semi-supervised machine learning algorithms fall somewhere in between supervised and unsupervised learning because labeled and unlabeled will be used to train the data.
- Reinforcement machine learning algorithms interacted with its environment by making actions and locates the errors. This method permitted the machines and software agents to involuntary determine ideal behavior within specific context to maximize their performances.

Machine learning authorize an analysis of huge amount of data. Besides, it mainly distributes the correct results faster to recognize the profitable chances or dangerous threat. In addition, it required additional time and resources to train correspondingly [22] By combining machine learning between AI with cognitive technologies can make it more productive in measuring information in large values.

During first research by [18] use machine learning and reverse engineering technique to detect the android malware detection. The authors focused on static approach based on an automatic analysis of decompiled mobile application codes. In this research, the unique feature vector derived from Java code application was build. There are 696 number of features. The authors divided them within three categories which are model implementation of onReceive() methods for BroadcastReceiver component. There are also commands group that obtain administrative access to the device, expand the opportunities of attack and hide the operation of malware on that devices. Based on the selected basis [5] the API Calls contains the largest group of features which is 616.

The second research by [2] also use the machine learning technique in their research to improve Android malware detection using big data of analytics. The author provided a comparison of seven different machine learning classifiers on the SherLock dataset [19] which is one of the largest datasets of Android malware. Using 35 GB of dataset and 17 node Spark cluster, the authors comparing the different classifiers including Logistic Regression, Isotonic Regression, Random Forest, Gradient Boosted Trees, Decision Trees, Support Vector Machine (SVM), and Multilayer Perceptron. Besides, they observed the tree based on techniques provide better result in general. Moreover, Gradient boosted trees provided approximately 91% precision and it is the highest among all the seven techniques. The authors also compared the FPR in detecting benign applications and observed that the gradient boosted tree techniques have lowest false positive. Furthermore, they deployed their trained model on private cloud to facilitate the malware application detection in real-time. Therefore, the authors envisioned that a service could be extremely useful for the communities.

In the third research by [17] the main purposes are to solve the problem about malware detection depends on the network monitoring instead using static approaches by analyzing the different network-based detection solutions that engaged the machine learning techniques and proposes enhancements to detect the malware precisely. The evaluation process consists two stages which are performing experiments and analysis. The experiment executes three sub stages such as data collection, machine learning classifiers and feature selection and extraction. During data collection stage, network traffic for benign and malicious applications was captured.

The existing studies of malware detection system used a dataset, features, precision and etc. to calculate either the malware is existed or not in the system. Despite that, none of the existing machine learning system can give permission to prevent the malware from entering their system. Therefore, this research would like to improve an existing system by giving permission to certain software that cleared from the malicious code in it. Meanwhile, for the software that contains malware in it, the system would not give any permission from entering the system.

secondary

3. RESEARCH METHOD

This section describes details techniques, approaches, and features applied during this research along with the methodology that will be used in carrying out the experiment. Fig.1 presented major components of malware detection system. There are three parts in these architectures which are collection of data, machine learning and the database itself. Each of phases are related to each other. The data collection process started by gather all permission of benign and malware applications. This process involves decompiling of .apk file, extract and filter permission. The permissions will be collected and store into a readable format as a .arff file. The a .arff file own every attributes of the features that will be used in features optimization approach to exclude the noise and irrelevance contains in the dataset [15].

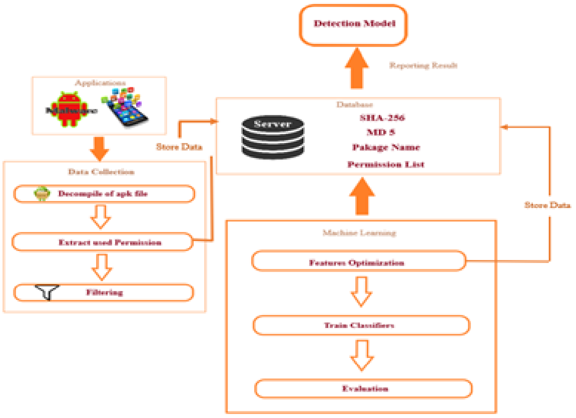

Figure 1: Android Malware Architecture

3.1 Data Collection Phase

The data collection process required benign and malware applications datasets in. apk file. During this phase, the random samples are drawn from AndroZoo datasets. The AndroZoo collected an executable of Android application from many sources and made the analysis available through it [16]. Besides, the dataset was confined to the applications that have been drawn from the Google Play. Furthermore, the Bouncer detection that have been implemented in Google Play are able to remove malware applications that bring harm towards their users. Therefore, by downloading the applications that came from Google Play is more accurate since it typically used to produce for the dataset of benign applications [11]. The processes of collection of data are illustrated as shown in Figure 2.

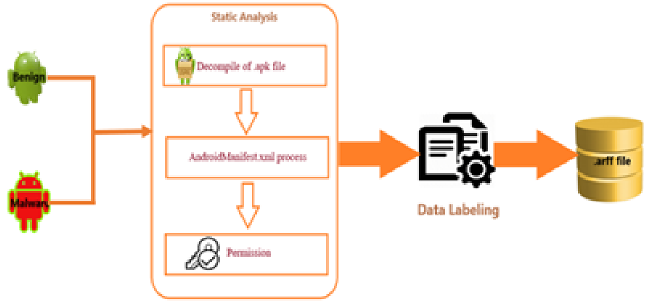

Figure 2: Data Collection Phase

3.2 Decompiling APK File

AndroidManifest.xml file uses to get essential information like permission and activities of application information. All the permission that has been extracted needs to be saved as x.arff file and loaded in WEKA. Meanwhile, the permission values stored as binary number (0 or 1). Moreover, the optimization feature is used to help in gaining the best features of permissions.

The permission features of malware detection were trained and classified by using significant features. This study applied the features selection to get an outstanding feature for the best malware detection. Features selection methods identified and removed unsuitable or unnecessary attributes data that cannot contribute towards any preciseness of predictive model [20] [23]. Hence, number of malware features was reduced from the Top 20 permission to Top 15 permission. This is to make sure there is a unique pattern between the benign and malware. This study also applied tenfold cross-validation approaches which being run frequently for ten times. Table 1 presented the permission features list that have been used by this study.

3.2 Machine Learning Classifier

Machine learning is artificial intelligence (AI) type that can learn without using explicit programming. It is capable of predicting future and improving decisions when revelead to new data. Prediction process usually based on the search through data set that look for patterns and referred as learning. The learning process and prediction results depends on its classifier types. This technique widely used to classify samples particularly in intrusion detection systems (benign and malware) area. The two commons type of machine learning are supervised and unsupervised [24]. Supervised machine learning approach have been applied in the research since the sample data set contains the labels (benign and malware).

Table 1: Attributes of Cleveland dataset

Abbreviation	Fullname
ACCESS_WIFI_STATE	Allow Wi-Fi networks information to be access by applications.
ADD_VOICEMAIL	Able any voicemail to be add by the applications into the system.
GET_ACCOUNTS	Allow the account list in the Accounts Service to be access.
GET_TASKS	Allow the app to recover information the current and new running tasks.
INSTALL_SHORTCUT	Allow shortcut to be install by application in Launcher.
MEDIA_CONTENT_CONTROL	Allow the applications to know the playing content and control its playback.
MOUNT_UNMOUNT_FILESYSTEMS	Allow mounting and unmounting files system for removable storage.
NFC	Allow I/O operations to be perfrom by I/O over the NFC.
READ_PHONE_STATE	Allow read only to phone state, phone number, any ongoing calls status, current cellular network information and all the list of Phone Accounts registered on the device.
SET_ALARM	Allow the Intent to be broadcast by application for alarm.
SYSTEM_ALERT_WINDOW	Allow the windows to be create by applications using type WindowManager.LayoutParams. TYPE_APPLICATION_OVERLAY on top of the other apps.
WRITE_EXTERNAL_STORAGE	Allow the external storage to be write from applications.
ACCESS_FINE_LOCATION	Allow the specific location to be access by applications.
ACCESS_COARSE_LOCATION	Allow the approximate location to be access by applications.
CHANGE_WIFI_STATE	Allow the changes of state of Wi-Fi connectivity by using applications.

Moreover, supervised machine learning offered fine result through error reduction. During research, the five classifiers have been implemented to perceive every particular results account in different types of machine learning classifiers. The five classifiers are Random Forest (RF), J48, Multi-Layer Perceptron (MLP), DecisionTable and Naïve Bayes.

Random Forest (RF): Well-known method of collective learning for supervised classification or regression. This machine learning technique worked by building random set of decision trees during training and producing the classes which mean the prediction (regression) of an individual trees or class mode (classification) [7].

MLP: Multi-layer perceptron is the artificial neural network model. It consisted multiple node layers that interacting through weighted connections [6].

J48: Known as ID3 extension. The additional features of J48 account for continuous range values, the trimming of decision trees, missing values and rules derivation. J48 is known as open source of Java in data mining tool Realization of the C4.5 algorithm (WEKA).

DecisionTable: Concise the visual representation to specify which actions need to conduct depends on its condition. The information conveyed the decision tables can represent as decision trees or a series if-then-else and switch-case statements of programming language.

Naïve Bayes: An algorithm that used Bayes' theorem to classify an object. Naive Bayes classifiers assumed naïve or strong independence between data points attribute. Naive Bayes classifiers is trendy for its text classification and traditional solution for spam detection problem.

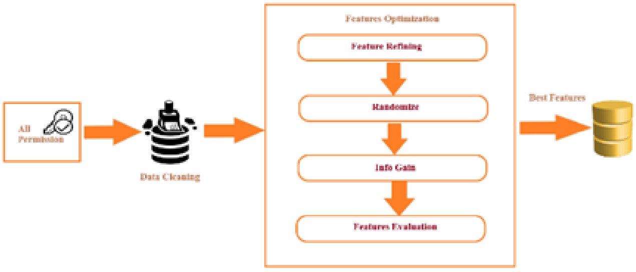

Figure 3: Machine Learning Phase

4. RESULTS AND ANALYSIS

This chapter will discuss about experiments that have been conducted for Android malware detection system using machine learning. Besides, initial outcome shows the results obtained from Random Forest, Naïve Bayes, J48, Decision Table and MLP. This study also used parameters of accuracy, FPR, precision, recall and f-measure to investigate the different measurements of each classifier. The outcomes achieved from 15 permission features of testing set which used five selected classifiers as in Table 2.

Table 2: Performance from each classifier

Classifiers	Accuracy (%)	FPR	Precision	Recall	F-measure
Random Forest	89.36	10.64	89.4	89.4	89.4
Naïve Bayes	86.03	13.97	86.0	86.0	86.0
J48	89.18	10.82	89.2	89.2	89.2
Decision Table	89.13	10.87	89.1	89.1	89.1
MLP	89.12	10.88	89.1	89.1	89.1

4.1 Comparative Analysis

The outcomes show Random Forest classifiers attained the accurateness result, 89.36% compared to Naïve Bayes which achieved only 89.2%. This determine that Random Forest classifiers is better than other classifiers in detecting the malware. The features selection also played a critical part in deciding the effectiveness of android malware detection. Fig.4 shows that the approach able to detect an unknown malware with over 89% rate of accuracy.

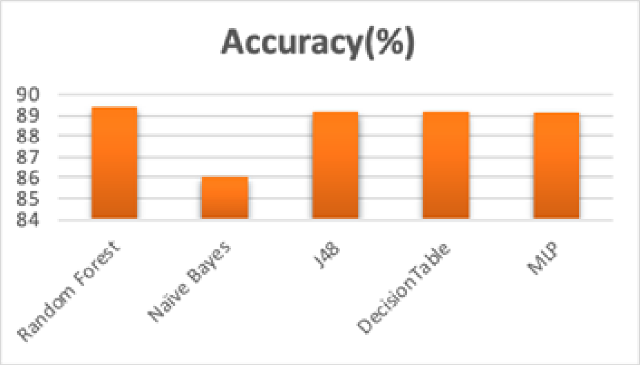

Figure 4: Percentage Accuracy

Confusion matrix technique summarized the classifications of model performance. Table below presented the possible classes of prediction, benign and malware. Like an example, if the model forecast the existence of any malware activities, the outcomes will appear as "malware" and vice versa. Therefore, five classifiers performance has been presented as Table 3.Table 3 expressed the study produced the best outcomes by predicting a malware of 8741 from J48 classifiers. Meanwhile incorrectly predicted perspective revealed that J48 got the slightest value. Therefore, J48 classifiers capable to predict malware more precisely. Based on the permission features in this study, the process is classified as benign and malware. Besides, the receiver operating characteristics (ROC) curve for each of machine learning classifiers also being calculated. During this stage, TPR is counted as detection rate which exactly predicted process of malware while FPR is choose as the detection rate which inaccurately predicted normal as malware. The curves of five machine learning classifiers have been presented as figure below.

Table 3: Confusion Matrix of classifiers

Classifiers	**Actual**	**Prediction**	
		Benign	**Malware**
Random Forest	Actual Benign	9159	841
	Actual Malware	1288	8712
Naïve Bayes	Actual Benign	9135	865
	Actual Malware	1929	8071
J48	Actual Benign	9095	905
	Actual Malware	1259	8741
Decision Table	Actual Benign	9126	874
	Actual Malware	1300	8700
MLP	Actual Benign	9100	900
	Actual Malware	1276	8724

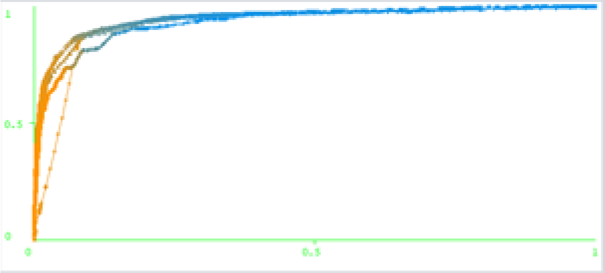

Figure 5: ROC Curve

The vertical axis in figure above shows the detection rate while horizontal axis shows error detection rate. Five lines represented ROC curve for each machine learning classifiers. Furthermore, ROC curve is not easy to compare due the similarities under the same conditions. Meanwhile, AUC used to calculate the accuracy of detection which resulted good or bad such as in table below. Therefore, area of 1 indicated the perfect prediction and area of 0.5 indicated a bad prediction.

Table 4: AUC results

Classifiers	**AUC**	**Prediction**
Random Forest	0.949	Perfect Prediction
Naïve Bayes	0.928	Perfect Prediction
J48	0.921	Perfect Prediction
DecisionTable	0.943	Perfect Prediction
MLP	0.948	Perfect Prediction

Table 5: Time taken to produce model (seconds) Classifiers Build model

Classifier	**AUC**
Random Forest	8.44
Naïve Bayes	0.18
J48	1.03
DecisionTable	2.89
MLP	30.43

Table 5 presented the time taken to produce results in second. The results show that Naïve Bayes has the lowest model complexity as it uses minimal time to build the model. Hence, it can be concluded that to achieve reasonable accuracy and effectiveness in classifying unknown malware as it helps to determine the performance of the classifiers.

After completing the experiments, the findings of the study show that machine learning able to produce the most accurate detection using five types of machine learning classifiers. The obtained results from previous research paper seems to be agreed that machine learning provides the best result in prediction. Based on this experiment, the result from analyzing the permission features of dataset provide better result compared to other result because using machine learning classifiers are more accurate to analyses and train the large of dataset. Based on the obtained result also it shows that Random Forest classifiers able to produce the most accurate detection result. Meanwhile, J48 also capable to produce result nearly with the target result but it is not accurate as Random Forest. Meanwhile the Decision Table and MLP produced the same result obtained. The less accurate data is Naïve Bayes.

5. CONCLUSION

A conclusion might elaborate on the importance of the work or suggest applications and extensions. As conclusion, this study has summarized in details about the development and methodology for Android malware detetion. The machine learning classifiers used in identifying the reliable features permission to get an accurate detection process during the process. Besides, this study appraised many machine learning classifiers types that able enhancing the performance of Android malware detection. During this study, the large training samples been extracted and successfully identified the better potential classifiers. Besides, 10k benign and 10k malware of data were being analyzed by five types of classifiers. It is very important to know the most suitable data that can leads to most accurate result in detect the malware using machine learning.

A set of real benign and Android malware data sample applications were used during this experiment. Moreover, static analysis technique act as classifier in order to differentiate data sample between benign and malware applications. Besides, the process of machine learning comprised three stages which are features optimization, trained classifiers and machine learning classifiers evaluation.

The outcomes indicated 89.36% detection rates of TPR using Random Forest classifiers on the samples of Drebin malware. The results obtained shows that using machine learning approach can reach high accuracy of detection rate and indicate its efficiency. This can prove that machine learning classifiers are capable in detecting the Android malware. In conclusion, the improvement of features optimization and learning classifiers can continuously derive to get the greatest obtain of detection performances.

There are several enhancements that can be carried out for future improvement of water level prediction by using machine learning such as False Alarms referred to the statistical measurement of how well the sample dataset classifies the Android malware correctly. This means that the malware data was incorrectly predicted as benign. This problem is lead to incorrect detection of malware and even small amounts of false alarms can cause huge impact. A reliable and efficient detection module is needed in order to solve this problem. Besides, the more complex and extensive data, the harder it becomes to choose the relevant features to improve detection performances. The process required further exploration to investigate the correlation between malware and benign applications. This will reduce False Alarms; hence it increases the detection accuracy. This study also can be done by using Dynamic Analysis approach. It able to identify the vulnerabilities during runtime environment.

ACKNOWLEDGEMENT

The work is funded by the Ministry of Higher Education FRGS Racer under Project ID: RACER/1/2019/ICT03/UMP//2 and UIC 190807.

REFERENCES

1. F.-X. Geiger and I. Malavolta, “Datasets of Android Applications: a Literature Review,” 2018, [Online]. Available: https://arxiv.org/pdf/1809.10069.pdf.
2. L. U. Memon, N. Z. Bawany, and J. A. Shamsi, “A comparison of machine learning techniques for android malware detection using apache spark,” J. Eng. Sci. Technol., vol. 14, no. 3, pp. 1572–1586, 2019.
3. “Machine Learning What it is and why it matters,” SAS, 2019..https://www.sas.com/it_it/insights/analytics/machine-learning.html
4. Feixiang He, Bohdan Melnykov, and Elena Root, “RottenSys: Not a Secure Wi-Fi Service At All,” Check Point Research, 2018.. https://research.checkpoint.com/2018/rottensys-not-secure-wi-fi-service/
5. S. Seo et al., “Mobile Malware Threats and Defenses for Homeland Security To cite this version : HAL Id : hal-01542454 Mobile Malware Threats and Defenses for Homeland Security,” 2017.
6. F. A. Narudin, A. Feizollah, N. B. Anuar, and A. Gani, “Evaluation of machine learning classifiers for mobile malware detection,” Soft Comput., vol. 20, no. 1, pp. 343–357, 2016.
7. F. Vanhoenshoven, G. Napoles, R. Falcon, K. Vanhoof, and M. Koppen, “Detecting malicious URLs using machine learning techniques,” 2016 IEEE Symp. Ser. Comput. Intell. SSCI 2016, no. December, 2017
8. Gregg Keizer, “Google reveals Android malware ‘Bouncer,’ scans all apps, Computerworld.,” Computerworld, 2012. .
9. S. Y. Yerima, S. Sezer, and I. Muttik, “High accuracy android malware detection using ensemble learning,” IE Inf. Secur., vol. 9, no. 6, pp. 313–320, 2015.
10. GDATA, “Cyber attacks on Android devices on the rise,” GDATA, 2018. .
11. M. F. A. Razak, N. B. Anuar, F. Othman, A. Firdaus, F. Afifi, and R. Salleh, “Bio-inspired for Features Optimization and Malware Detection,” Arab. J. Sci. Eng., vol. 43, no. 12, pp. 6963–6979, 2018.
12. Symantec, “Internet Security Threat Report,” 2018.
13. Statista, “Global market share held by the leading smartphone operating systems in sales to end users from 1st quarter 2009 to 2nd quarter 2018,” Statista, 2019.

.https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/

14. D. Palmer, "FalseGuide malware dupes 600,000 Android users into joining botnet," ZDNet, 2017.
15. S. Y. Yerima, S. Sezer, and G. McWilliams, "Analysis of Bayesian classification-based approaches for Android malware detection," IET Inf. Secur., vol. 8, no. 1, pp. 25–36, 2014.
16. Z. Mas'Ud, S. Sahib, M. F. Abdollah, S. R. Selamat, and R. Yusof, "Analysis of features selection and machine learning classifier in android malware detection," ICISA 2014 - 2014 5th Int. Conf. Inf. Sci. Appl., pp. 1–5, 2014.
17. D. A. Alotaibi, M. F. Aldakheel, N. S. Al-serhani, R. Zagrouba, I. Technology, and S. Arabia, "MACHINE LEARNING BASED ON MALWARE DETECTION IN MOBILE," vol. 31, no. 3, pp. 505–511, 2019.
18. M. Kedziora, P. Gawin, and M. Szczepanik, "Android Malware Detection Using Machine Learning And Reverse Engineering," no. 616, pp. 95–107, 2018.
19. Y. Mirsky, A. Shabtai, L. Rokach, B. Shapira, and Y. Elovici, "SherLock vs moriarty: A smartphone dataset for cybersecurity research," AISec 2016 - Proc. 2016 ACM Work. Artif. Intell. Secur. co-located with CCS 2016, pp. 1–12, 2016,
20. T. S. Chou, J. Pickard, and C. Popoviciu, "Machine learning based IP network traffic classification using feature significance analysis," WMSCI 2018 - 22nd World Multi-Conference Syst. Cyber Informatics, Proc., vol. 1, no. 3, pp. 1–3, 2018.
21. E. S. Team, "What is Machine Learning? A definition," Expert System, 2017
22. N. S. Zaini et al., "Phishing detection system using machine learning classifiers," Indones. J. Electr. Eng. Comput. Sci., vol. 17, no. 3, pp. 1165–1171, 2019,
23. M. F. A. Razak, N. B. Anuar, R. Salleh, A. Firdaus, M. Faiz, and H. S. Alamri, "'Less Give More': Evaluate and zoning Android applications," Meas. J. Int. Meas. Confed., vol. 133, pp. 396–411, 2019.
24. M. H. Kamarudin, C. Maple, T. Watson, and N. S. Safa, "A LogitBoost-based Algorithm for Detecting Known and Unknown Web Attacks," IEEE Access, vol. 3536, pp. 1–12, 2017.
25. O. V. Lee et al., "A malicious URLs detection system using optimization and machine learning classifiers," Indones. J. Electr. Eng. Comput. Sci., vol. 17, no. 3, pp. 1210–1214, 2020.
26. N. S. Zaini, D. Stiawan, A. F. Mohd Faizal Ab Razak, S. K. Wan Isni Sofiah Wan Din, and T. Sutikno, "Phishing detection system using machine learning classifiers," Indones. J. Electr. Eng. Comput. Sci., vol. 17, no. 3, pp. 1165–1171, 2020.