International Journal of Innovative Research in Advanced Engineering (IJIRAE) ISSN: 2349-2163
Issue 09, Volume 8 (September 2021) https://www.ijirae.com/archives

COMPARATIVE ANALYSIS OF MACHINE LEARNING ALGORITHMS FOR THE DETECTION OF ANDROID MALWARE

Efefiong Udo-Nya
Cyber Security Science, Federal University of Technology, Minna. Niger State, Nigeria
efefiongudonya@gmail.com

Olawale Surajudeen Adebayo
Cyber Security Science, Federal University of Technology, Minna. Niger State, Nigeria
waleadebayo@futminna.edu.ng

Publication History
Manuscript Reference No: IJIRAE/RS/Vol.08/Issue09/SPAE10084
Research Article | Open Access
Peer-review: Double-blind Peer-reviewed
Article ID: IJIRAE/RS/Vol.08/Issue09/SPAE10084
Received: 22, September 2021
Accepted: 30, September 2021
Published Online: 01, October 2021
Volume 2021 | Article ID SPAE10084 | https://doi.org/10.26562/ijirae.2021.v0809.004
Citation: Efefiong & Olawale (2021). COMPARATIVE ANALYSIS OF MACHINE LEARNING ALGORITHMS FOR THE DETECTION OF ANDROID MALWARE. International Journal of Innovative Research in Advanced Engineering, VIII, 265-271. **doi:** https://doi.org/10.26562/ijirae.2021.v0809.004
Editor-Chief: Dr.A.Arul Lawrence Selvakumar, Chief Editor, IJIRAE, AM Publications, India

Abstract: This paper examines the effectiveness of some machine learning algorithms in the detection of android malicious application. In order to carry out this analysis, drebin dataset of android malicious and good applications were obtained and used for the classification as described in a section of this article. The classification results show that the Cubic SVM, Quadratic SVM and ensemble Subspace KNN performed better with 99.2%, 98.7% and 98.4% accuracy with 0.0079, 0.0129 and 0.1598 error rate respectively.

Keywords: Android Platform, Machine Learning, Classification, Malware, SVM, Ensemble Method, Mobile Device

I. INTRODUCTION

Smartphones remains one of the most popular technologies in high demand, due to its ubiquitous nature as a result of its adaptable functionalities and diverse usage. Their activities and usage have permeated into different facets and spheres of life, with an increasingly widening scope of acceptance having become a necessity to human in the present modern/digital world. Mobile platforms are almost indispensable in this technological era due to its wide range of enhanced functionalities, and has far outpaced PCs, laptops. In fact, there is no gainsaying that the advent and sophistication of mobile technology has enhanced better efficiency and effectiveness of human life and activity.
The increasing functionalities on smartphones to allow optimal exploration or use, has increased the complexity of its platform, thus making it prone to software vulnerabilities and several other inherent risks. Consequently, this has drawn the attention of hackers and cybercriminals, culminating in the violation of users' privacy in barrage of ways, unauthorised access and privileges, and the disruption and destruction of information, information systems and critical infrastructures.

It is not an understatement to say that the ubiquitous nature of and advancement of cutting-edge technology via Internet of things (IoT), and proliferation of IT devices and infrastructures like smart phones, computers, tablets, have created the leeway and a quite conducive environment and platform for creation and proliferation of malware, which has propelled exponential growth in different levels of cyber threats and cyber-attacks. Android platform, due to its openness, easy access, and ease of operability, is considered a leader with an overwhelming market share in the world of smartphones OS, having gained or acquired superb or very large market share on billions of smart devices used around the world. It remains one of the most patronised or used mobile platform in the present digital age. Malwares are malicious software or applications designed to target OSs, computer systems, or network infrastructures, and some have specifically designed purpose, like the mobile Android malwares, tailored to attack mobile Android platforms. Interestingly, software vulnerabilities on Android smartphones, whose exploitation triggers unusual or unexpected behaviours in the system, is undoubtedly on the increase and quite challenging to detect or identify, due to complexity of the smartphone platforms.

Despite the security architecture and mechanism embedded in android devices, which restricts the apps to some environments and privileges, its OS is reported as the most targeted and affected mobile OS by malware threats [11]. Cybercriminals carry out these acts by injecting these malwares into the devices using different strategies. Detection of the malware can be achieved by studying the behaviours and malicious patterns of the processes [17]. It is expedient to note that while anti-malware developers are creating solutions to contain and avert existing malwares, malware developers on their part are re-strategizing and enhancing new and sophisticated techniques, like the code obfuscation, to help them perpetrate their nefarious activities, thus making these two sets of persons to constantly be at their heels to outsmart the other. Several losses have been incurred by individuals and organisations in different industries, thus motivating the unrelenting concern in the research on the security of Android platform. Suffice it to say that though different research efforts with remarkable success to combat this menace have been made using the Machine learning classification techniques alongside others, new Android malware vectors are still emerging, thus the need for continuous research in this domain space. Different ML classifiers have been trained with different datasets and methodologies to detect Android malwares, with various levels of successes, based on the methodologies and the strengths of the ML algorithms.

In this research paper, comparative analysis and evaluation of different machine learning classification techniques to detect Android malware on android platform using the DREBIN dataset was carried out. The drebin dataset has both the malware and benign data with 215 features from the API calls and system permissions. The remaining sections of the paper is organised as follows: section two discusses the related literatures to this research while section three describes the methodology which includes the data description, technology used, classification (training, validation and testing), section four encapsulates the result and analysis which include the measurement and evaluation metrics, and performance results, and section five covers the conclusion.

II. RELATED WORKS

Chavan et al (2019) carried out a research on the comparative analysis which covered the classification of both binary and multi-class family of android malware and benign, based on static features. They used Android malware Genome project dataset which consist of apk files from various malware forums and Android apps. The researchers dealt with the challenge of high feature dimensionality using the information gain approach and RFE based on a linear SVM to reduce the features of the dataset. Subsequently, they deployed some machine learning classifiers (random forest, ANN, linear svm. J48, LMT, random tree, and Adaboost on each of the resulting data subsets for malware classification and detection, alongside the cross-validation strategy. Evaluation and comparison of the created models was made using the precision and AUC. For information gain approach, the precision for linear SVM, J48, ANN, LMT Random tree and Adaboost are 0.96, 0.96, 0.97, 0.96, 0.97, and 0.96 respectively, The AUC for them in the information gained approach remain 0.94, 0.96, 0.96, 0.97, 0.97, and 0.99 respectively. Similarly, the precision for linear svm and J48 using the RFE with svm feature selection are 0.96 and 0.96.

Khan et. al. (2017) proposed a method of malware detection using the interceptor that sits in between the web browser and the server. They reduced the dimensionality of the datasets through wrapper method of feature selection where a sizeable feature subset was obtained and deployed for classification. Dataset used for the research work had 1924 instances of 409 and 1515 malicious and benign JavaScript respectively. They had 3 different experiments with different levels of partitions. In experiment 1, the entire subset of the dataset after feature selection was used for the training, whereas, in experiment 2 and 3, the data was splited into 80% training and 20% testing; and 10-fold cross-validation respectively. The following machine learning classifiers were engaged in the research work – Naïve Bayes, KNN, svm, and J48. SVM in all the three experiments achieved accuracy of 94.55% and 95.42%, compared to other classifiers like Naïve Bayes which had 95.06% and 97.99%; J48 with 99.22% and 98.64%, in the second and third experiments respectively. Yang et. al (2020) proposed an approach which is based on decision tree (DT) with support vector machine (svm) algorithm (DT-SVM) to improve the accuracy of classification. Their DT-SVM machine learning advanced algorithm extracted the Dalvik opcode of sample using the reversing Android software, the n-gram model was used to generate the eigenvectors of the sample. They generated a decision tree by training the sample and subsequently updated the decision nodes as svm nodes from bottom up. This research deployed the strength of both the DT and SVM especially overfitting reduction by svm, to have high accuracy. They asserted that they designed an Android malware defection framework which is based on the DT-SVM advanced algorithm.The work achieved an all-time precision of 96% using the DT-SVM algorithm, for the Android malware apps classification/detection with a relatively low time consumption. Wang et al (2019) postulated a robust Android malware detection approach based on selective ensemble learning capable. The study projected the SEdroid, which is an Android malware detection engine that is quite robust and engages the approach of selective ensemble learning and genetic algorithm.

SEdroid adopts the concept of comparative experiment, to showcase “a more robust and preeminent capability”, and upon evaluation produced the performance with the precision of 98.3% and 98.1% malware for recall ratio. The research posits that designing SEdroid with consideration to diversity of the ensemble and accuracy, facilitate and fast-track the process of finding optimal ensemble combination, thus providing the model with super robustness and very strong generalization ability.

Malware is a malicious software designed and implemented by hackers/attackers to meet the harmful or malicious intent or to carry out certain nefarious activities, which include to spread itself and remain undetectable, steals, cause changes or damages to confidential information, disrupt or gain unauthorised and fraudulent access to users' devices and networks, crippling of critical information systems and infrastructures, and infecting or compromising systems or networks. Mobile malwares are specifically written to attack mobile devices such as smartphones, tablets, smartwatches and other wearable devices. It explores and exploits vulnerabilities of the mobile OS and phone technology. Malware is a growing and one of the biggest and toughest threats to mobile devices, information systems and the internet at large. Each malware operates in a bewildering variety of forms with different attack vectors. Over 350,000 new malicious programs, are discovered and registered daily by the AV-TEST Institute. There is a tremendously and significant increase in the amount and variety of mobile malware programs that is targeting smartphones and tablet, and the growth rate is highly alarming [1]. The sophistication of malware attacks has increased at the emergence of "file-less" malware as an effective alternative form of attack. The non-requirement of an executable file in the endpoints for a malware execution, and the absence of footprint make it even more challenging to be detected [3]. In 2016, there was an alarming wave of WannaCry ransomware threat which attacked millions of computers across the globe. Mobile malware symptoms include unwarranted behaviours; degradation of device performance; stability issues such as frozen apps, failure to reboot and difficulty connecting to network, depleting battery life, reduced processing power, hijack of the browser, sending unauthorised SMS messages, and freezing or brinking the device [2]. There exist different types of mobile malware variation with varying attack vectors, different methods of distribution and infection, and impacts on mobile devices [4].

A. Classification of Malware

1) Worms: Worms are malicious programs which upon installing itself into the computer memory, replicate and infect the entire device or network. It spreads very fast through software vulnerabilities or phishing attacks. It can replicate and overwhelm the system resources like bandwidth consumption and server overload, delete files, and install a backdoor for unauthorised access. The type of worm and possibly, the security measures on the device or network determine the nature of harm it can perpetrate. Cybercriminals can transmit worms through Short Message Service (SMS) or Multimedia Messaging Service (MMS) text messages, and typically do not require user interaction to execute commands.

2) Virus: Virus is a malicious program which operates by attaching itself to an executable file, program or OS. It activates and spreads through the system at the launch of the executable program that it is attached to. Mobile viruses are adapted for the cellular environment and designed to spread from vulnerable mobile device to another. It can spread through websites, file sharing, email attachment downloads, and other downloads from unreliable websites. A computer or mobile virus can hijack applications, use these applications to send out infected files to other systems, clients, or contacts. Malicious parties can potentially use mobile virus to root the device and gain access to files and flash memory.

3) Bots and Botnets: A bot is an IT device like computers or mobile device that is infected with malware such that it can be controlled remotely by a hacker or cybercriminal and could be used to launch cyber-attacks. A collection of these bots also referred to as zombie, form a botnet which is limitless as they spread undetected. Hackers through the master servant commands, use the botnets to carry out several malicious activities including sending spam and phishing messages; keylogging screenshots and webcam access; DDOS attacks; and spreading other types of malware. These programs can operate in the background on the user device, concealing themselves and lying wait for certain behaviours like online banking session to strike. Hidden processes can execute completely, run executables or contact botmasters for new instruction, and still remain invisible to the user.

4) Trojan Horses: Trojan horse is a malicious program that disguises itself as a legitimate and trustworthy file or program and often activated by the users. Mobile trojan finds itself into devices by attaching itself to seemingly harmless or legitimate programs and get installed alongside with the apps after which it will infect the device or perpetrate malicious actions. Cybercriminals typically embed Trojans into non-malicious executable files or apps in the mobile devices. Trojans can infect and deactivate other applications and the mobile device itself as soon as it is activated. It can also paralyse the device after a certain period of time or a certain number of operations. These malicious programs hijack the browser, captures user login information from other apps such as mobile banking apps. Trojans themselves are a doorway. It can spy on devices or systems; capture or steal data; delete or modify data; harvest devices and make it part of botnet; and gain unauthorised access to devices and networks. Banking Trojans target vulnerable users by distributing fake version of legitimate mobile apps.

5) Ransomware: Ransom ware is a malicious program that locks the data on a victim's device typically by encryption, thus restricting devices or users access to their hardware devices, files or data, with a demand for a payment of ransom which most times are made with cryptocurrencies such as bitcoin; before the data or device is decrypted and access returned to the victim. In a ransomware attack, notice is often displayed on the device and instructions provided on how to recover the encrypted item. In May 2017, a ransomware named WannaCry attacked and compromised over 200k computers within just one day, spread across over 150 countries.

This attacked individuals and corporate bodies with monumental damages and losses estimated in the hundreds of million and billions of dollars. WannaCry affected Microsoft OS that did not have the latest patch installed for a known vulnerability.

6) Adware and Scams: These are malwares that automatically deliver advertisements, with irrelevant and unsolicited pop-ups and illegal ads to the users. Apart from posing as nuisance, adware can slow down user's device and also redirect it to malicious sites. A device compromised with adware can deliver spywares, which most often are easily hacked, thus making the devices to be soft target for hackers, phishers, and scammers. Most adware is authored by advertising firms as a revenue generating tool. Though some adware is designed to deliver advertisements, it is not uncommon for some of them to be bundled with spyware that is capable of tracking user activities and stealing personal and confidential information.

7) Spyware: Malicious program with a common threat which secretly keep records of all activities of the users (both online and offline), harvest the users' data and collect personal and confidential information such as contacts, usernames, passwords, location, downloads, user preferences, messaging habits, browser history and surfing habits/ behaviour and relays these data to a third party. It can also collect device information like International Mobile Subscriber Identity (IMSI) number, product ID, International Mobile Equipment Identity (IMEI) number, and OS version, which can be used by the third party to launch future cyber-attacks. Spyware is often installed or distributed on user device without the user's consent as a freeware or shareware with a disguised or appealing function at the front end as a legitimate app, with covert, nefarious and unknown mission running in the background. This means is often use for perpetrating identity theft and credit card fraud. Spyware at times are referred to as adware because they may be advertisers or marketing firms. Cybercriminals or advertisers have access to users' data through spyware and some of them can further install additional malware that make changes to the settings of devices.

B. Machine Learning

Machine learning is a technique that provides systems' ability to autonomously make decisions from a set of provided data, without any external support. ML makes such decisions by first learning from the data and further understanding the data patterns. Supervised learning, is used for data modelling where there is a precise mapping between input and output data. The algorithm for supervised learning has the capacity to recognise and identify the relationships between the two variables in order to have a prediction for a new outcome. Classification is the process of recognising, understanding and grouping ideas and objects into pre-set categories or sub-population. It is the process of deploying algorithm which use pattern recognition in training dataset in order to spot the various patterns which could be number sequences, sentiments or similar words in future or new datasets, and further make predictions on the likelihood of a subsequent dataset falling into predetermined categories using the training data. In fact, the main goal of a classification problem is to identify the category or class to which a new data will fall under. Binary classification is a classification task with two possible outcomes for instance gender classification. Multi-class classification is one with more than two classes. In multi-class classification, each sample is assigned to one and only one target label. Multi-label classification is a classification task where each sample is mapped to a set of target labels (more than one class). For instance, a news article can be about sports, a person, and a location at the same time. There are several types of classification algorithms and their usage depends on a dataset [18].

III. METHODOLOGY

A. Dataset Description

This research work used the drebin dataset that contain families of android malwares. The dataset has feature vectors of 215 attributes with 15,036 observations. The dataset was extracted from Drebin project with 5,560 malware apps and another 9,476 benign apps, all totalling 15,036 applications. Invariably, the dataset has 15,036 rows and 215 columns or features, in addition to the class label whose entries is either malware or benign. The dataset has a supporting file that has the description of the feature vectors (attributes) from two categories of API calls and system permission. In order to have a balanced dataset to work with, we used the 5560 malware instances alongside the 5561 benign that was randomly selected from the 9476.

B. Technology Used

MathWorks technology (matlab 9.6) was used for all implementations and evaluations.

C. Classification (Training, validation and testing)

In this section of the research work, we first partitioned the dataset into a training and validation sets, and applied the 10-fold cross-validation method to have training and test sets in each split.

The following classification algorithms RUSboosted trees (ensemble), subspace KNN (ensemble), subspace discriminant (ensemble), KNN weighted, cubic SVM, linear SVM, fine decision tree, linear discriminant, logistic regression, boosted trees (ensemble), SVM quadratic, and fine KNN, bagged trees (ensemble) and cubic SVM were trained using the training dataset and further tested or scored on the test set. A classification models emerged from the various trainings which were used during the testing phase to classify the test dataset.

IV. RESULTS AND ANALYSIS

A. Measurement and Evaluation Metrices

This metrics evaluates the model's performance which tells how good or bad the classification is. Each of the evaluation metric evaluates the model in different ways.

1) **Confusion Matrix:** This is a matrix representation with tabular visualisation of the real classification labels and the model predictions which measures the performance of the classifier.

Table I Confusion Matrix with Two Class Labels

Actual	Predicted	
	Positive	Negative
Positive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	True Negative (TN)

- ❖ True Positive (TP) – An instance in which the predicted and the real (actual) values are positive (true).
- ❖ True Negative (TN) - An instance in which the predicted and the real (actual) values are negative (false).
- ❖ False Positive (FP) – An instance in which the predicted value is positive whereas the real (actual) value is negative (false).
- ❖ False Negative (FN) – An instance in which the predicted value is negative (false) whereas the real (actual) value is positive (true).
- ❖ The type-1 error is equivalent False Positive, type- II is equivalent to False Negative.

The diagonal elements represent the number of points in which the predicted labels are equal to the true labels, whereas every other thing outside the diagonal were misclassified or mislabelled. The higher the diagonal values of the confusion matrix, the better.

2) **Accuracy:** This is the ratio of the number of the correct or right predictions to all the total predictions. So, it tells how the classifier often make correct prediction. Accuracy is put to use when the number of samples belonging to each class are equal.

$$\text{Accuracy} = (TP + TN) / \text{Total number of predictions}$$

3) **Misclassification Rate (Error Rate)**: This is a measure of the failure rate in terms of classification by the classifier.

$$\text{Misclassification Rate} = (FP+FN) / \text{total of predictions}$$

4) **Precision:** This stipulates the ratio of right or positive predictions to overall actual positive prediction. Precision is often put to use when there is a class samples imbalance

$$\text{Precision} = TP/(TP+FP)$$

5) **Recall or Sensitivity:** This metric is a measure of the true positive rate (TPR), which is a ratio of true positives to everything positive. Recall provides a better way of evaluating model performance in the face of a class imbalance. It is quite easy to compute recall, but it requires a threshold.

$$\text{Recall (TPR)} = TP/\text{Actual positive}$$

6) **F1_Score:** This is the harmonic mean of the precision and recall, where an F1_score reaches its best value at 1 (perfect precision and recall) and worst at 0.

$$F1\_Score = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$$

7) **Receiver Operating Characteristics (ROC) Curve:** This measurement is done by plotting the true positive rate versus the false positive rate. This plot produces the ROC curve, which allows the model designer to visualise the trade-off between the true positive rate and the false positive rate.

8) **False Alarm Rate:** This can be considered as false positive rate (FPR) or false discovery rate (FDR). But for this research, we will use the FDR

$$FDR = FP/(FP + TP)$$

9) **Error Rate:** This is the ratio of the total number of misclassifications to the total number of predictions.

$$\text{Error Rate} = (FN + FP)/(P+N)$$

B. PERFORMANCE RESULT

Table II. Result of comparative performance of the Classifiers

Algorithms	Accuracy %	False Alarm rate	Precision	Error rate	Recall	F1_Score	Prediction Speed (obs/sec)	Training Time (sec)
RUSboosted Trees	92.4	0.0726	0.927	0.0759	0.929	0.9269	9800	354.83
Decision Tree (Fine tree)	95.0	0.0407	0.959	0.049	0.945	0.9516	21000	36.921
Subspace KNN (Ensemble)	98.4	0.0189	0.981	0.159	0.9885	0.9847	25	1921.1
Linear SVM	97.4	0.0266	0.973	0.0259	0.977	0.9749	9800	57.792

Subspace Discriminant (Ensemble)	96.7	0.0329	0.969	0.0329	0.967	0.9679	2600	236.46
KNN weighted	98.1	0.0247	0.975	0.0237	0.979	0.9769	220	603.33
Linear Discriminant	95.8	0.033	0.967	0.0419	0.952	0.9593	10000	134.98
Cubic SVM	99.2	0.0095	0.9904	0.0079	0.9942	0.9923	8800	39.582
Boosted Trees (Ensemble)	96.3	0.0235	0.976	0.0369	0.952	0.9636	14000	94.417
Logistic Regression	97.6	0.0265	0.973	0.0239	0.981	0.9769	7400	406.09
Quadratic SVM	98.7	0.0114	0.9885	0.0129	0.9866	0.9875	11000	33.81
KNN Fine	97.4	0.0266	0.973	0.0297	0.977	0.9749	330	181.57
Bagged Trees (Ensemble)	98.2	0.0153	0.985	0.0179	0.981	0.9829	6600	125.78

C. RESULT DISCUSSION

The comparative analysis of the performances of the various classifiers in the table above shows that the Cubic SVM, Quadratic SVM, Subspace KNN (ensemble) classifiers clearly outperforms other classifiers, with the Cubic SVM being adjudged as the best based on its accuracy rate of 99.2 percent, and false alarm rate, precision, error rate, recall and f1_score of 0.0095, 0.9904, 0.0079, 0.9942, and 0.99229 respectively. The RUSboosted trees (ensemble) and the fine decision tree had the least performance, with accuracies of 92.4 percent and 95.0 percent, though their performance still falls within an acceptable level. Though not listed in the of classifiers, it was realised that the Naïve Bayes classifier performed very poorly as it failed to even converge.

IV. CONCLUSION

There is no doubt that the surge in the android malware is still in the increase thus portraying the importance of the research in the detection of android malware. This research work was carried out using the API features calls and system permission of android applications which encompassed 215 features or attributes. Pertinent to mention that almost all the classifiers performed within the acceptable performance rate, except the Naïve Bayes which did not converge. Cubic SVMhad an excellent accuracy, precision and recall rates which demonstrated effective efforts in detecting android malwares. This classifier also demonstrated low false positive rate though the reverse of this is not as expensive as having the false negative rate. Further efforts, can be deployed in optimising these classifiers to obtain more precise and accurate rates. In addition, more techniques can be explored to enhance the detection of android malwares.

REFERENCES

1. Adebayo O. S. and Normaziah Abdul Aziz (2019). Improved Malware Detection Model with Apriori Association Rule and Particle Swarm Optimization. Security and Communication Networks.
2. Adebayo O. S, Joel Anyam, Shefiu Ganiyu, Sule Ajiboye Salawu (2020). Analysis and Classification of some Selected Social media Apps Vulnerability. Springer. Book collection of International conference on Information and Communication. Part of the Communications in Computer and Information Science book series (CCIS, vol. 1350, page 456 - 469.
3. Adebayo O. S and Aziz N.A (2015). Static Code Analysis of Permission-based Features for Android Malware Classification Using Apriori Algorithm with Particle Swarm Optimization. Journal of Information Assurance and Security, Vol. 10 (4). ISSN 1554-1010. Available at www.mirlabs.net/jias/index.htm.
4. Adebayo O. S., Aziz N. A. (2014). Android Malware Classification Using Static Code Analysis and Apriori Algorithm Improved with particle swarm optimization. 4th World Congress on Information and Communication Technologies, Malaca, Malaysia. Review link: https://easychair.org/conferences/submission.cgi?submission=2010970;a=7484291
5. Adebayo O. S, Aziz N. A. (2014). Techniques for Analysing Android Malware. The 5th International Conference on Information and Communication Technology for the Muslim World (ICT4M), 2014, Sarawak Malaysia. Review link: https://easychair.org/conferences/submission.cgi?a=6897964;submission=1902969
6. Aneja, L., Babbar, S. (2018) Research trends in malware detection on Android devices. Springer Nature, (799), 629–642,https://doi:10.1007/978-981-10-8527-7_53
7. Bhatia, T., Kaushal, R. (2016). Malware Detection in Android based on Dynamic Analysis,
8. Coronado-De-Alba, L. D., Rodríguez-Moto, A., Escamilla- Ambrosio, P. J. (2016). Feature Selection and Ensemble of Classifiers for Android Malware Detection, Proceedings of 2016 IEEE International Conference
9. Chavan, N., Troia, F. D., Stamp, M. (2019). A comparative analysis of Android malware.
10. Dubey, A., Misra, I. (2013). Android security attacks and defenses, Parkway NW: CRC Press Taylor & Francis
11. Elenkov, N. (2015). Android Security Internals, An In-Depth Guide to Android's Security Architecture, San Francesco: William Pollock

12. Gibert, D., Mateu, C., Planes, J. (2020). The rise of machine learning for detection and classification of malware: Research developments, trends and challenges, Journal of Network and Computer Applications, (153), https://doi:10.1016/j.jnca.2019.102526

13. Jami, Q., Shah, M.(2016). Analysis of Machine Learning Solutions to Detect Malware in Android, 2016 INTECH Innovation Conference

14. Khan, N., Abdullahi, J., Khan, A. S. (2017), Defending malicious script attacks using machine learning classifiers. Wireless Communications and Mobile computing, (2017), https://doi:/10.1155/2017/5360472

15. Memon, L. U., Bawany, N. Z., Shamsi, J. A. (2019). A comparison of machine learning techniques for Android malware detection using Apache Spark, Journal of Engineering Science and Technology, 14(3), 1572 - 1586

16. Ng, A. P.. Chiew, K. L., Ibrahim, D. H. A., Tiong, W. K., Sze, S. N., Musa, N.(2018) Android malware detection technique via feature analysis, Journal of Engineering Science and Technology, (2018) 78 – 90

17. Martin, I., Hermader, J. A., Munoz A. (2018). Android characteristic using metadata machine https://doi:10.1155/2018/5749481

18. Rathore, H., Agarwal, S., Sahay, S. K., Sewak, M. (2019). Malware detection using machine learning and deep learning

19. Ranveer, S., Hiray, S., (2015). SVM Based Effective Malware Detection System. International Journal of Computer Science and Information Technologies, Vol. 6 (4), 2015, 3361-3365

20. Wen, L., Yu, H. (2017). An Android malware detection system based on machine learning. AIP Conference Proceedings, (1864), https://doi:10.1063/1.4992953

21. Wang, J., Jing, Q., Gao, J., & Qiu, X. (2020). SEdroid: A Robust Android Malware Detector using Selective Ensemble Learning. 2020 IEEE Wireless Communications and Networking Conference (WCNC). https://doi:10.1109/wcnc45663.2020.912053

22. Yang, M., Chen, X., Zhang, H., (2020). An Android malware detection model based on DT-SVM. Security and Communication Networks, (2020), https://doi:10.1155/2020/8841233

