4th International Engineering Conference (IEC 2022)
Federal University of Technology, Minna, Nigeria

Systematic Literature Review on Android Malware Detection

* Anyaora, P. C[1], Adebayo, O. S[2], Ismalia, I[3], Ojeniyi, J. A[4] & Olalere, M.[5]
[1]Cyber Security Science Department, Federal University of Technology, PMB 65 Minna Niger State, Nigeria
[2]Department of Computer Science, Islamic University in Uganda, P.O Box 2555, Mbale, Uganda
*Corresponding author email: P.anyaora@futminna.edu.ng +2348139319538

ABSTRACT
Users of Android-powered smartphones and tablets have multiplied dramatically. Thanks to Android third-party apps, the essential applications, such as banking and healthcare, are accessible on Android smartphones. There are new threats to be taken into account about harmful programs when these applications are utilized and embraced more broadly. This research performs a systematic literature review using the prima framework and Kitchenham statement to apply on android malware detection and analysis of different methodology of publishing research that have been used for android malware detection for the last past five years. Using the keyword" Android malware detection" the research had seen over 610 published articles on" Android Malware detection". It was narrowed down to 142 published research papers due to it between the year 2018 to 2022 that was looked at, sixty-five articles (65) were finally selected for investigation after inclusion and exclusion. One of the research key findings is the performance of Machine Learning (ML) algorithms which were relatively higher than others.

Keywords: Android malware detection, dynamic analysis, static analysis, machine learning algorithm.

.

1 INTRODUCTION

In terms of market share, Android smart phones currently make up more than 80% of all smart phones, and by 2020, analysts estimated that they will reach 85%. A record-breaking increase in the number of new android applications has been brought on by the increasing popularity of android smartphones, and these apps have also attracted the attention of hackers. By the end of 2018, there were 856.52 million different types of Android malware in existence. There were around 137.5 million new malicious applications found in 2018 (or 350,000 new viruses every day) (Y. Zhang et al., 2020). There are several operating systems available for mobile devices. For smartphones and tablets, the most popular operating system is Android, which is free and open-source. Google claims that 1.3 million android smartphones are activated every day. (2016) Arshad et al. The development of Android technology has drawn many malware authors. In order to make money from the production of malware programs, malicious authors are constantly honing their craft. Such programs could directly violate the security of the Android operating system. As a result, the victim's financial credentials and personal information are at danger. Android device malware attacks have reached a crucial stage. It is anticipated that as the usage of smartphones grows, malware dangers will increase. Malware-containing applications are many and risk the security of the Android operating system (OS) (Fan et al., 2020).Malware types including Trojans, phishing software, spyware, and other types are used by Android malware on mobile devices.

Malware makers repackage and distribute popular Google Play apps on other third party app stores to take advantage of program vulnerabilities; as a result, both app merchants and developers suffer. These harmful software programs, such as viruses, Trojan horses, adware, back doors, or spyware, can infiltrate mobile devices and disrupt or harm the operating system while stealing confidential data. To get beyond Android's antimalware safeguards, malware writers employ code obfuscation, dynamic execution, stealth, encryption, and repackaging (Felt et al., 2011). A detailed knowledge of harmful applications is required to stop such infections so that proper security measures to safeguard user data may be adopted (Felt et al., 2011). There are some advantages in combining several classifiers such as increasing robustness, obtaining better accuracy, and heavily built generalization. Understanding malware in it various forms and studying various techniques in which malware can be reduced to the barest minimum is a necessity in dealing with malware in android's smart phones. there have also been novel innovations developed that has helped the curbing of malicious applications. A presented understanding of various types of datasets is necessary, Techniques (Adebayo & Aziz, 2014) are discussed in relation to the study of these attack vectors in order to find and collect important information for analysis, categorization, and recommendations for the best solution.

Benefits of each malware categorization method were clearly emphasized in another study (Olawale Surajudeen Adebayo et al., 2012). The study outlined the many types of malware, malware categorization methods, malware behaviors, and techniques for avoiding and eradicating malware, should it ever infiltrate a system. The research

the tools that identify malware datasets using a rule-based classification scheme and machine learning algorithms are also described. These tools use pattern recognition to distinguish harmful programs from legitimate programs. It is imperative to stop the malicious effects of malware since they pose a global danger to our internet resources and financial transactions. The portions of this document listed below are arranged as follows: The literature review, which includes the connected works, is provided in section 2, the research methodology is described in part 3, the findings and conclusions of the review are presented in section 4, and the study is completed in section 5. The figure 1: shows the global statistics of smartphone sales from 2007 to 2021. These sales are in millions, thus it is clear that sales of android phones have been extremely successful from the moment they were produced.

2 RELATED WORKS

All nearest Neighbors (ANN), Weighted All Neighbors (WANN), First Nearest Neighbors (FNN), and K-medoid Four malware detection techniques, based Nearest Neighbors (KMNN), employ Hamming distance to find similarities between samples. (Taheri and others, 2020) Their recommended solutions permit the activation of an alarm if they judge an Android app to be hazardous. Therefore, their techniques help to stop malware detection from disseminating extensively. Their research demonstrates that the proposed algorithms' accuracy rates are higher than 90% and, in some cases (such as when taking API features into account), higher than 99%, and are comparable to the most recent state-of-the-art solutions. Taheri et al. (2020) combined the static and dynamic analysis (hybrid analysis). The research of (Alzaylaee et al., 2019) suggest DL-Droid, a deep learning system that uses stateful input generation and dynamic analysis to find malicious Android applications. On actual devices, over 30,000 experiments with both benign and malicious applications were conducted. Additionally, tests were performed to compare the stateful input generation method's detection performance and code coverage to the standard stateless approach using the deep learning system.

According to their research, DL-Droid outperforms conventional machine learning algorithms and can detect android malware with detection rates of up to 97.8% with dynamic features alone and 99.6% with dynamic features combined**.**(Shatnawi et al., 2022) research got an insight, examined the effectiveness of four machine learning classifiers that aim to identify malware based on both static (permissions) and dynamic (action repetition) features. These characteristics were discovered to have a significant impact and play a crucial part in the classification procedure. they specifically used four classifiers in three steps. They made use of the dataset's dynamic properties in the initial step. In the second stage, static features were used, and in the third stage, a mix of static and dynamic characteristics were used.

Additionally, the research of (Zhu et al., 2022) proposes a hybrid deep network learning technique called Stacked Hybrid Learning. By including a more conventional deep learning method called Stacked De-Noising Auto-encoders, MSAE and SDAE (SHLMD) are established. To further improve the capability of detecting malware, more compact and discriminative characteristics are extracted from the rich features (SDAE). They trained a malware detection model utilizing the feature learned by the MSAE and SHLMD, respectively, using classification approaches as Support Vector Machine (SVM) or K-Nearest Neighbor (KNN). Results from tests on two real-world datasets show that SHLMD achieves accuracy rates of 94.46% and 90.57%, respectively.

Then uniquely, a novel Apriori association rule with better malware detection model was suggested by Adebayo and Aziz (2019). During an unsupervised learning experiment, it had a root mean square error of 0.0355, an average accuracy of 98.17%, a detection rate of 98.25%, a false alarm rate of 0.0192, and a false alarm rate of 0.0192. Additionally, the memory and temporal complexity of the new model show increased computational effectiveness. It was investigated how the Apriori method may be improved for the extraction and selection of candidate detectors for classifier training. Particle swarm optimization was used to enhance the Apriori approach and boost its efficiency in producing candidate detectors for supervised learning. A hybrid strategy of machine learning and genetic algorithms is reportedly being offered by study (Srinivasan et al., 2022) for identifying Android malware. This is a strong and effective solution. They reviewed the Android system architecture, security features, and malware classification in their brief analysis of Android applications. Machine learning-based algorithms are used to identify malware more successfully if signature-based techniques are unable to detect novel varieties of malware that pose zero-day risks.

Additionally, the study by Gao et al. (2021), which employed a hybrid analytic technique on Android malware in their research and the currently accessible datasets, which only offer a rigid and constrained picture of Android malware over a short period of time, have disregarded the evolving nature of Android malware. In general, the time variable has never been properly considered, ignoring idea drift. Additionally, it has been ignored where dynamic data comes from and what makes it unique. The time and data platform source issues must be resolved in order to create more reliable, strong, and durable detection systems. Combining data from several data sources for both benign and dangerous software over a longer time period yielded 489 static and dynamic

characteristics for their study. Using an actual device and an emulator to access dynamic data sources (such as system calls), two equally-featured datasets were produced in order to take into consideration the features of different types of devices (Gao et al., 2021). According to the Artificial Malware-based Detection (AMD) dynamic detection technique proposed by, Android Malware Detection uses both malware patterns that have been collected and those that have been artificially built (Jerbi et al., 2020). The fake malware patterns are produced by an evolutionary (genetic) process. With the latter, a population of API calls is made in an effort to identify various malware behaviors using a specified set of evolution criteria. The created fraudulent activities are then put to the examples base to expand it with fresh malware patterns. The major goal of the proposed AMD method is to increase the rate of detection by varying the pool of malware instances (Jerbi et al., 2020). Additionally (Iqbal & Zulkernine, 2019) proposed a Spy Droid which employs the dynamic analysis, a framework for real-time malware detection that may support a number of detectors from outside sources (such as researchers and antivirus providers) and permits effective and in-depth real-time monitoring. Spy Droid supports application layer in addition to two operating system modules, sub-detectors (monitoring and detection). Sub-detectors are typical Android apps that communicate their findings to the detection module after using the monitoring module to track and assess various runtime data. The detection module chooses when to flag a program as malicious. It demonstrates how choices made by a number of sub-detectors may significantly increase the malware detection rate on a real device. A novel slow-aging strategy dubbed SDAC (using dynamic analysis) is proposed by (J. Xu et al., 2022) to address the problem of model aging in Android malware detection by not responding to changes in Android specifications during malware detection. The API call sequences that were gathered from Android apps are used to evaluate the contexts of the APIs. The differences between the API vectors are viewed as the semantic distances, and a neural network is applied to the sequences in order to assign APIs to vectors. SDAC then creates feature sets by clustering all APIs based on their semantic distances, which it then expands in the detecting phase to incorporate all new APIs. Without requiring to be trained by a fresh set of real-labelled apps, SDAC may adapt to changes in Android standards by just identifying new APIs that appear during the detection phase. Comprehensive experiments utilizing datasets with dates ranging from 2011 to 2016 show that SDAC achieves a significantly higher accuracy and a notably slower aging speed. The dynamic analysis is also used in the research of (J. Zhou et al. 2020) to create a traffic fingerprint by analyzing the characteristics of malicious traffic on the host machine. A viable detection method that is appropriate for encrypted communications is created by combining machine learning techniques. An extra layer called a confusion classifier is implemented to aid with malware classification in order to distinguish between identical fuzzy traffic. They replicated two situations for classification using a real-world dataset named CICAndMal2017: malware binary detection and malware category categorization. The testing findings indicate that while the accuracy rate for malware category categorization is 95.2%, it is 98.8% accurate for detecting malware binary. Applying the dynamic approach, the research of (Kumar and Thomas 2021) suggests a brand-new behavioral strategy for Android malware detection and categorization as another dynamic analytic methodology. In the suggested method, the dataset of Android malware is decompiled to find the suspect API classes and methods and to provide an encoded list. Using the encoded patterns, the multiple sequence alignment for various malware families is produced, it is then used to produce a profile hidden Markov model. Based on the resulting log likelihood score, the model determines whether an unknown program is malicious or benign. In comparison to other current frameworks for the detection of Android malware, the framework's accuracy of 94.5% is noticeably greater.

In their study (Elayan and Mustafa 2021), characteristics were found using static analysis. Employing a Gated Recurrent Unit (GRU), a kind of Recurrent Neural Network, they offered a fresh technique for spotting malware in Android applications (RNN). Permissions and calls to the Application Programming Interface (API) are the two static components that they retrieved from Android apps. The CICAndMal2017 dataset is used to evaluate and train their methodology. The test results show that their deep learning system performs better than the competitors with an accuracy of 98.2%. Additionally, H. Zhou et al. (2020) provide an Android-based SIMGRU-based static malware detection approach. We utilize similarity to enhance the Gated Recurrent Unit (GRU) and produce three separate SimGRU structures: InputSimGRU, HiddenSimGRU, and InputHiddenSimGRU. This is because the similarity of clustering is commonly employed in static analysis of Android malware. InputHiddenSimGRU is produced by combining HiddenSimGRU with InputSimGRU. According to their experiment, InputSimGRU, HiddenSimGRU, and InputHiddenSimGRU outperform the regular GRU model and other methods (Elayan & Mustafa, 2021). The research of (Ibrahim et al., 2022) provided a novel approach by using static analysis to compile the two most recently proposed features as well as the most advantageous components of Android applications. They then fed this data into a practical API deep learning model they created. The method was used to analyze a brand-new and labeled collection of Android application samples, which included 14079 malware and benign samples split into four distinct malware categories.

This dataset was utilized in two significant studies, the first of which focused on the detection of malware using samples from the dataset separated into only two groups—malware and benign—and the second on the detection and classification of malware using samples from all five classes in the dataset. A 99.5% F1 score was achieved using just two courses (İbrahim et al., 2022).

The research of (Sandeep et al. 2019) also recommended using a static analysis technique in conjunction with a fully connected deep learning model to detect Android malware. One of the main features of the work is the identification of Android malware, version packages, and detection of Android malware even before installation. Additionally, it has an extraordinary 94.65% accuracy rating. This model also learnt every feature from every conceivable combination of features. It was put through extensive testing and research to achieve exceptional accuracy. Additionally, according to the findings of the research (Ndatsu, Z. & Adebayo, 2020), models with selected permission-based features are more accurate than models without feature selection. Additionally, the research of (Alswaina & Elleithy, 2018) used a static analysis technique and machine learning to assess and identify malware attributes such as the permissions sought by malware. In their research, they concentrated on identifying a limited subset of permissions that may be used to categorize programs into the correct malware families. They further decreased the number of features (by 0.28%) from 59 to 42 using Extremely Randomized Trees. Their two methods, they expressed the chosen characteristics as weighted values (MWcand) and as binary values (MBcand). With KN over StormDroid, they were able to increase accuracy by 0.02% (RF, 95.99%) and reduce time performance by 37.5%. when they assessed their methods using the accuracy and time performance of six classifiers (Alswaina & Elleithy, 2018). Also the research of (Raghav et al., 2021) examined how current machine learning and deep learning algorithms for detecting android malware make use of different feature building techniques. Most of these feature development techniques make use of frequency-based vectors made from various files included in the Android application bundle (APK). The semantic information that is included in those files is not preserved by these approaches for creating features based on frequency. In order to create feature vectors that can accurately represent the data found in the android manifests and dalvik executable files present inside an APK, they (Raghav et al., 2021) proposed a method that makes use of the static analysis and document embedding natural language processing (NLP) technique. These embeddings are then used to build binary classifiers that can accurately distinguish between a good and bad Android application.

3 METHODOLOGY

For the search period from January 1st, 2018 to November 2022, this review utilized the Preferred Reporting Items for Systematic Reviews and Meta-Analysis (PRISMA) (Page et al., 2021). Research questions, a research method, and selection criteria were all used to achieve the study's goal. Figure three (3) show us statistics on the trends on proffering android malware detecting techniques to detect android malwares and even other mobile devices, there have been significant progress in proffering solutions over the years due to more malwares are been created by programmers. The figure three (3) also shows the statistics from 2016 to 2028, prediction on how researchers will proffer more solution in future.

3.1 **Phases of the Study and the Protocol, Section**

In conducting this review, the Preferred Reporting Items for SLRs and Meta-Analyses (PRISMA) statement (Page et al., 2021) was adhered to. The established principles from the study of (Kitchenham et al., 2009) were also utilized to apply the SLR to the field of computer science.

3.2 **Study's eligibility requirements and exclusions**

The study used a five-point criterion to determine if a research paper was eligible to be chosen for the review. Table 1 displays the standards and pertinent justifications.

3.3 **Information Sources and the Search Process:**

For the manual search, the following databases were used: IEEE Xplore Digital Library, Elsevier, Research Gate, and Google Scholar. In order to find new and valuable resources for the study, the search terms were manually searched across the chosen internet databases and sources. These are some of the essential phrases: android malware detection" and "systematic literature review on android malware detection. Figure four (4) using the Prisma framework as our methodology to explain the inclusion and exclusion of different research publication on android malware detection gotten from different research publication databased used. At the identification stage the systematic literature review identified 142 after duplicate was excluded and the date range which it had worked 2018 to 2022, IEEE explore (55), research gate (19), Google scholar (54), Elsevier (14), then at the screening stage forty-four (44) articles was screened out from one hundred and forty-two (142) articles and left with ninety-eight (98) articles. At the eligibility stage, articles which were eligible for the SLR considering the date range of 5 years, the key word" android malware detection" and all other SLR were excluded left with ninety-eight (98), also the SLR research work was able to get seventy-nine (79) full research publication excluding nineteen (19) publication further.

The research also had to still narrow the research article to sixty-five (65) due to six (6) of the article were not well explained, six (6) of the article cannot detect android
malwares, seven (7) of the articles were not research articles, sixty-five full article was finally selected with no inclusion from using reference follow up or additional records obtained using personal contact.

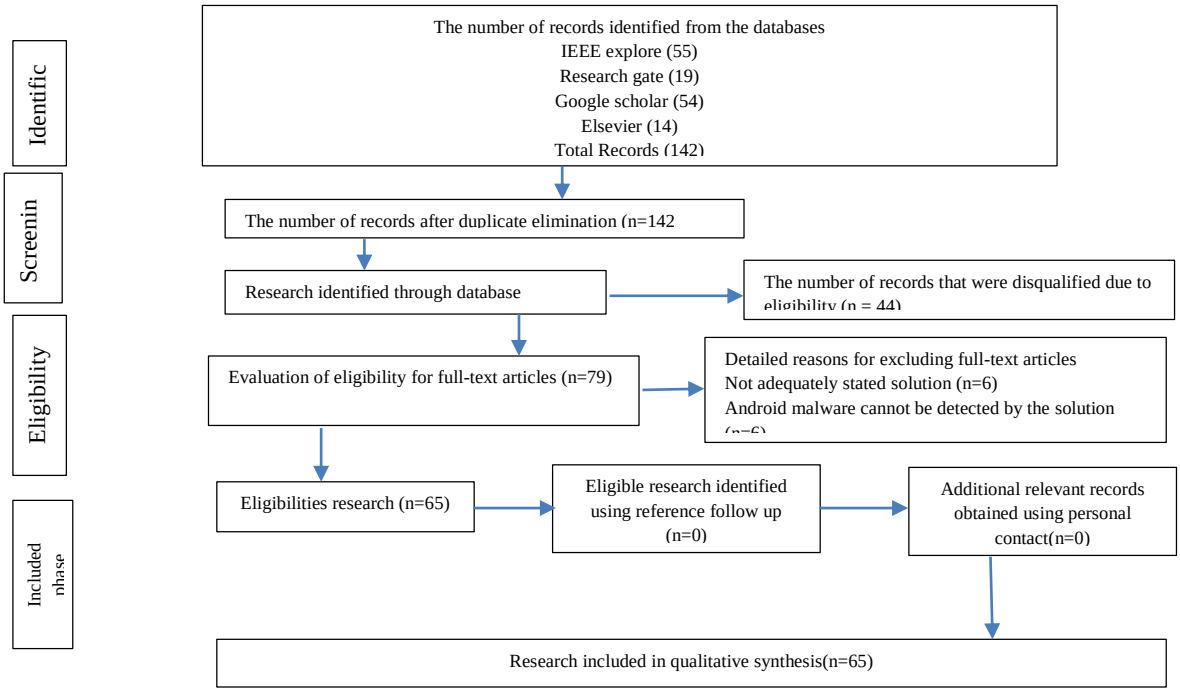

Figure 4: The study selection workflow with Prisma Framework.

3.3 Study selection and data collection processes

All of the titles and abstracts of the pertinent papers that were retrieved from various databases and sources were independently checked for eligibility. The appropriate items were gathered after careful deliberation. Using the titles, abstracts, and pertinent data gleaned by the researcher, further searches were made for full-text papers containing possibly pertinent studies. A Microsoft Excel template was used to compile the retrieved data. The research articles' data that have been extracted include:

1. The specifics of the study, such as the primary author and the year of publication.
2. The process used to extract features from the Android application.
3. The method(s) employed for android malware detection or prevention.
5. The technique's precision or success rate.

Table 1: Criteria for the Article Inclusion and Exclusion of Research Publications summarizes the various factors and justifies the criteria that were taken into account when choosing which research articles to publish in this systematic literature review.

SN	Criteria	Explanation/Justification
1.	A publication of original research, not a review or survey,	The research articles should cover anti-malware strategies, malware, and/or malware tactics, including how to use them and how they operate.
2	The offered solutions must be	The goal of this study is to offer recommendations to security

	capable of malware detection or prevention.	software developers and policy makers for the development of safer work practices and systems.
3	The article must be a whole paper.	Short papers fall short in presenting essential details about the suggested fix.
4	The articles must be written in English as the language of choice.	English must be used in the publication.
5	The article has to be released between 2018 and 2022.	The SLR covers the years 2018 through 2022.

ensemble technique, then the Identification of analysis technique which include static and dynamic analysis.

4 RESULTS AND DISCUSSIONS.

This section relays the outcomes of the systematic reviews carried out in this research. The literatures were able to categorized malware and their techniques according to the following subsections: Top ten malware family, its behavior and purpose on android devices: table 2 list top ten of recent malwares family, also the behavioral pattern and it purpose on android devices. Also it was able to Identify different kinds of features for classification like permission based and API and identifying the need for applying machine learning algorithm for selection of features which helps the machine learning algorithm to perform better during classification. The literature was able to clearly identify lots of classification techniques like machine learning, use of association rule analysis (alternative method),

Table 2:Different malwares, behaviour and purpose.

Name of family	Behavior	Purposes
FakeInt	Trojan	Sending premium rate SMS messages
OpFake	Trojan, Ad-ware	Sending premium rate SMS messages and deceiving the target in terms of his/her browser being out of date through the use of pop-up messages.
SNDApps	Trojan	Sending private information to the server without the user knowing
Boxer	Trojan	Sending premium rate SMS messages
GinMaster	Trojan	Storing the victim's private information, such as mobile ID, mobile number, and other important
VDDLoader	Trojan	Flooding devices in terms of messages and sending private information to remote server
FakeDolphin	worm	Deceiving the victim by mimicking the dolphin browser and then signing up subjects without their consent and redirecting them when they browse to websites where the FakeDolphin is downloaded
Basebridge	Trojan	Sending premium rate SMS messages and blocking data consumption monitoring.
Kungfu	Backdoor	Gaining unauthorized access to the victim's devices, downloading a malicious app package, and sending the stored information from the memory of the device

		to the server.
JIFake	Trojan	Sending premium rate SMS messages, gathering personal information, and tracking location data.

5 Conclusion

The systematic review of the literature (SLR) on Android malware detection, prevention, and issues served as the foundation for this work. The study is being offered to give readers, internet users, and security managers a better grasp of Android malware mediums and vectors, strategies, and anti-malware methods. Anti-malware strategies for Android have been looked at, analyzed, and assessed for this reason. It was discovered some of the top ten most recent Android malwares during the SLR study, including GinMaster, Fake Dolphin, Kungfu, JIFake, SNDApps, OPFake, FakeInt, Basebridge, VDDLoader, and Boxer. It is encouraged to employ machine learning algorithm to combat android malware. This study's primary goal is to provide a comprehensive understanding of current Android malware and anti-Android malware techniques. Future research on Android malware detection is anticipated to focus more on dynamic and hybrid analysis, as there has been a noticeable lack of work on dynamic analysis. Researchers are also anticipated to create more ways to extract more recent features of static and dynamic analysis. The Appendix A table below shows all the authors of the full sixty-five (65) articles used for the systematic literature review, the accuracy achieved and method that was used. As you peruse the table, you will realized that all the authors made use of machine learning algorithm and significant high accuracy for the solution proposed in their research work, this will enable researchers work on the existing achievement and develop more models using machine learning to be able detect recent android malwares.

APPENDIX A

Authors	DOI	Techniques	Accuracy
1. (Taheri et al., 2020)	10.1016/j.future.2019.11.034	machine learning	90%
2. (S. Wang et al., 2020)	10.1016/j.ins.2019.11.008	machine learning	0
3. (Jerbi et al., 2020)	10.1016/j.cose.2020.101743	machine learning	0
4.(Alzaylaee et al., 2020)	10.1016/j.cose.2019.101663	machine learning	99.6%
5.(Ren et al., 2020)	https://doi.org/10.1016/j.adhoc.2020.102098	machine learning	93%
6.(Elayan & Mustafa, 2021)	https://doi.org/10.1016/j.procs.2021.03.106	machine learning	98.2%
7.(H. Zhou et al., 2020)	10.1109/ACCESS.2020.3007571	machine learning	0
8.(H. Han et al., 2020)	10.1109/BigComp48618.2020.00-96	machine learning	99.75%
9.(K,2020)	10.1109@ICOEI48184.2020.9142929	machine learning	96.46%
10. (Zhu et al., 2021)	10.1109/TNSE.2020.2996379	machine learning	89.07%
11.(Shatnawi et al., 2022)	https://doi.org/10.1155/2022/1830201	machine learning	0
12.(Iqbal & Zulkernine, 2019)	10.1109/MALWARE.2018.8659365	machine learning	94%

13. (Zhu et al., 2022)	10.1109/TKDE.2021.3067658	machine learning	
14.(Z. Xu et al., 2021)	10.1109/CSCloud-EdgeCom52276.2021.00021	machine learning	0
15.(İbrahim et al., 2022)	10.1109/ACCESS.2022.3219047	machine learning	96%
16. (Alani & Awad, 2022)	10.1109/ACCESS.2022.3189645	machine learning	98%
Authors	**DOI**	**Techniques**	**Accuracy**
17. (Ndatsu, Z. & Adebayo, 2020.)		machine learning	0
18.(J. Zhang et al., 2021.)		machine learning	0
19.(Rathore, 2021)		machine learning	0
20.(Shyong et al., 2020)		machine learning	96%
21.(Srinivasan et al., 2022)	10.1088/1742-6596/2325/1/012058	machine learning	0
22.(Srivastava et al., 2020)	10.1109/SMART50582.2020.9337105	machine learning	0
23. (Sandeep, 2019)	10.1109/ICCS45141.2019.9065765	machine learning	94.65%
24.(Geremias et al., 2022)	10.1109/IWCMC55113.2022.9824985	machine learning	0
25.(Rathore, 2021)	10.1109/PERCOMWORKSHOPS51409.2021.9430980	machine learning	0
26. (J. Xu et al., 2022)	10.1109/TDSC.2020.3005088	machine learning	97.49%
27.(Yuan et al., 2021)	10.1109/TSMC.2019.2958382	machine learning	0
28. (J. Zhou et al., 2020)	10.1109/ICCWAMTIP51612.2020.9317429	machine learning	98%
29. (N. Zhang et al., 2021)	10.1016/j.asoc.2020.107069	machine learning	0
30.(Billah et al., 2018)	10.1016/j.diin.2018.01.007	machine learning	0
32.(Sihag et al., 2021)	10.22667/JISIS.2021.05.31.034	machine learning	0
33. (Jerbi et al., 2020)	10.1016/j.cose.2020.101743	machine learning	0
34.(Kumar & Thomas, 2021)	10.1016/j.pmcj.2021.101336	machine learning	94.5%
35.(Gao et al., 2021)	10.1016/j.cose.2021.102264	machine learning	97%
36. (Guerra-manzanares et al., 2021)	10.1016/j.cose.2021.102399	machine learning	0
37.(Ding et al., 2020)	10.1007/s12652-020-02196-4	machine learning	95.1%
38. (X. Wang & Li, 2021)	10.1016/j.neucom.2020.12.088	machine learning	94%
39. (Cai et al., 2021)	10.1016/j.neucom.2020.10.054	machine learning	0
40. (W. Wang et al., 2018)	10.1007/s12652-018-0803-6	machine learning	0
41. (Cavallaro & Goos, 2021)	https://doi.org/10.1007/978-3-030-80825-9	machine learning	0

42.(Şahİn et al., 2022)	10.1109/ACCESS.2022.3146363	machine learning	0
43. (Ri et al., 2021)	10.1109/UBMK52708.2021.9558983	machine learning	94%
44. (Chandok et al., 2022)	10.1109/INCET54531.2022.9824877	machine learning	0
45. (Song, 2021)	10.1109/TRUSTCOM53373.2021.00115	machine learning	98.57%
46. (Dener et al., 2022)	https://doi.org/10.3390/app12178604	machine learning	99.97%
47. (Adebayo & Aziz, 2019)	https://doi.org/10.1155/2019/2850932	machine learning	0
48.(Y. C. Chen et al., 2021)	10.1109/ACCESS.2021.3110408	machine learning	0
49. (Alswaina & Elleithy, 2018)	10.1109/ACCESS.2018.2883975	machine learning	95.99%
50.(H. Zhang et al., 2021)	10.1109/DSC53577.2021.00100	machine learning	0
51.(Haq et al., 2021)	10.1109/ACCESS.2020. 3079370	machine learning	0
52.(Gohari et al., 2021)	10.1109/ICWR51868.2021.9443025	machine learning	97.29%
53. (Hashem & Fiky, 2021)	10.1109/MIUCC52538.2021.9447661	machine learning	0
54. (Raghav et al., 2021)	10.1109/ICDMW53433.2021.00104	machine learning	83%
55. (Feng et al., 2020)	10.1109/ACCESS.2020.3008081	machine learning	95.22%
56. (K, 2020)	machine learning	93.46%	
Authors	**DOI**	**Techniques**	**Accuracy**
57. (M. Chen & Wang, 2022)	10.1109/ICAICA54878.2022.9844642	machine learning	97.47%
58. (Hadiprakoso et al., 2020)	10.1109/ICOIACT50329.2020.9332066	machine learning	99.08%
59. (Sharma & Agrawal, 2022)	10.1109/CSNT54456.2022.9787671	machine learning	94.92 %,
60.(Parker et al., 2019)	10.1109/MALWARE.2018.8659372	machine learning	79%
61. (Q. Han et al., 2020)	10.1109/TIFS.2020.2975932	machine learning	0
62. (Y. Chen & Chen, 2021.)	10.1109/DSC49826.2021.9346277	machine learning	0
63. (Gong et al., 2021)	10.1109/TMC.2021.3079433	machine learning	97%
64.(Faisal Ahmed et al., 2022)	10.1109/DASA54658.2022.9764984	machine learning	97.5%
65. (Bayazit et al., 2022)	10.1109/HORA55278.2022.9800057	machine learning	98.85%

REFERENCE

• *Cell phone sales worldwide | Statista*. (n.d.). Retrieved October 1, 2021, from https://www.statista.com/statistics/263437/global-smartphone-sales-to-end-users-since-2007/

7 Types of Computer Malware and How to Prevent Them in 2022 - TitanFile. (n.d.). Retrieved December 2, 2022, from https://www.titanfile.com/blog/types-of-computer-malware/Adebayo, O. S., & Aziz, N. A. (2014). Techniques for analysing Android malware. *2014 the 5th International Conference on Information and Communication Technology for the Muslim World, ICT4M 2014*, 1–6. https://doi.org/10.1109/ICT4M.2014.7020656

Adebayo, O. S., & Aziz, N. A. (2019). Improved Malware Detection Model with Apriori Association Rule and Particle Swarm Optimization. *Security and Communication Networks*, *2019*. https://doi.org/10.1155/2019/2850932

Alani, M. M., & Awad, A. I. (2022). PAIRED: An Explainable Lightweight Android Malware Detection System. *IEEE Access*, *10*(June), 73214–73228. https://doi.org/10.1109/ACCESS.2022.3189645

Alsobeihy, M., Altamimi, S., Salem, E., Alhazzani, H., & Alhjaile, E. (2020). Using Machine Learning to Classify Android Application Behavior. *2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering, CSDE 2020*, 12–15. https://doi.org/10.1109/CSDE50874.2020.9411630

Alswaina, F., & Elleithy, K. (2018). Android Malware Permission-Based Multi-Class Classification Using Extremely Randomized Trees. *IEEE Access*, *6*(December), 76217–76227. https://doi.org/10.1109/ACCESS.2018.2883975

Alzaylaee, M. K., Yerima, S. Y., & Sezer, S. (2020). *Computers & Security DL-Droid : Deep learning based android malware detection using real devices*. *89*. https://doi.org/10.1016/j.cose.2019.101663

Alzubaidi, A. 2021. (2021). Recent Advances in Android Mobile Malware Detection: A Systematic Literature Review. *IEEE Access*, *(Volume:*(2169–3536), 146318–146349. https://doi.org/10.1109/ACCESS.2021.3123187

Arshad, S., Ali, M., Khan, A., & Ahmed, M. (2016). Android Malware Detection & Protection: A Survey. *International Journal of Advanced Computer Science and Applications*, *7*(2). https://doi.org/10.14569/ijacsa.2016.070262

Bayazit, E. C., Sahingoz, O. K., & Dogan, B. (2022). A Deep Learning Based Android Malware Detection System with Static Analysis. *HORA 2022 - 4th International Congress on Human-Computer Interaction, Optimization and Robotic Applications, Proceedings*. https://doi.org/10.1109/HORA55278.2022.9800057

Billah, E., Debbabi, M., Derhab, A., & Mouheb, D. (2018). MalDozer : Automatic framework for android malware detection using deep learning. *Digital Investigation*, *24*, S48–S59. https://doi.org/10.1016/j.diin.2018.01.007

Cai, M., Jiang, Y., Gao, C., Li, H., & Yuan, W. (2021). Neurocomputing Learning features from enhanced function call graphs for Android malware detection. *Neurocomputing*, *423*, 301–307. https://doi.org/10.1016/j.neucom.2020.10.054

Cavallaro, L., & Goos, G. (2021). *Detection of Intrusions and Malware , and*.

Chandok, A., Verma, A., & Gupta, R. (2022). *Dro-Mal Detector : A Novel Method of Android Malware Detection*. 1–9.

Chen, M., & Wang, K. (2022). *An Android Malware Detection Method Using Deep Learning based on Multi-features*. 187–190. https://doi.org/10.1109/ICAICA54878.2022.9844642

Chen, Y. (2020). *Android malware detection system*

integrating block feature extraction and multi-head attention mechanism. 408–413. https://doi.org/10.1109/ICS51289.2020.00087

Chen, Y. C., Chen, H. Y., Takahashi, T., Sun, B., & Lin, T. N. (2021). Impact of Code Deobfuscation and Feature Interaction in Android Malware Detection. *IEEE Access*, *9*, 123208–123219. https://doi.org/10.1109/ACCESS.2021.3110408

Chen, Y., & Chen, G. (2021). *Using Generative Adversarial Networks for Data Augmentation in Android Malware Detection*. https://doi.org/10.1109/DSC49826.2021.9346277

Dener, M., Ok, G., & Orman, A. (2022). *applied sciences Malware Detection Using Memory Analysis Data in Big Data Environment*.

Dhalaria, M. (2021). *Android Malware Detection using Chi-Square Feature Selection and Ensemble Learning Method*. 36–41.

Ding, Y., Zhang, X., Hu, J., & Xu, W. (2020). Android malware detection method based on bytecode image. *Journal of Ambient Intelligence and Humanized Computing*, *0123456789*. https://doi.org/10.1007/s12652-020-02196-4

Elayan, O. N., & Mustafa, A. M. (2021). Android malware detection using deep learning. *Procedia Computer Science*, *184*(January), 847–852. https://doi.org/10.1016/j.procs.2021.03.106

Faisal Ahmed, M., Tasnim Biash, Z., Raihan Shakil, A., Ann Noor Ryen, A., Hossain, A., Bin Ashraf, F., & Iqbal Hossain, M. (2022). ShielDroid: A Hybrid Approach Integrating Machine and Deep Learning for Android Malware Detection. *2022 International Conference on Decision Aid Sciences and Applications, DASA 2022*, 911–916. https://doi.org/10.1109/DASA54658.2022.9764984

Fan, M., Liu, T., Liu, J., Luo, X., Yu, L., & Guan, X. (2020). Android malware detection: a survey. In *Scientia Sinica Informationis* (Vol. 50, Issue 8). Springer International Publishing. https://doi.org/10.1360/SSI-2019-0149

Felt, A. P., Finifter, M., Chin, E., Hanna, S., & Wagner, D. (2011). P3-Felt. *Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, 3–14.

Feng, J., Shen, L., Chen, Z., Wang, Y., & Li, H. U. I. (2020). *A Two-Layer Deep Learning Method for Android Malware Detection Using Network Traffic*. *8*(July 2013).

Gao, H., Cheng, S., & Zhang, W. (2021). TC 11 Briefing Papers GDroid : Android malware detection and classification with graph convolutional network. *Computers & Security*, *106*, 102264. https://doi.org/10.1016/j.cose.2021.102264

Geremias, J., Viegas, E. K., Santin, A. O., Britto, A., & Horchulhack, P. (2022). *Towards Multi-view Android Malware Detection Through Image-based Deep Learning*. 572–577. https://doi.org/10.1109/IWCMC55113.2022.9824985

Gohari, M., Hashemi, S., & Abdi, L. (2021). *Android Malware Detection and Classification Based on Network Traffic Using Deep Learning*. 71–77. https://doi.org/10.1109/ICWR51868.2021.9443025

Gong, L., Li, Z., Wang, H., Lin, H., Ma, X., Liu, Y., & Ieee, F. (2021). *Overlay-based Android Malware Detection at Market Scales : Systematically Adapting to the New Technological Landscape*. *1233*(c), 1–12. https://doi.org/10.1109/TMC.2021.3079433

Guerra-manzanares, A., Bahsi, H., & Nõmm, S. (2021). KronoDroid : Time-based Hybrid-featured Dataset for Effective Android Malware Detection and Characterization. *Computers & Security*, *110*, 102399. https://doi.org/10.1016/j.cose.2021.102399

Hadiprakoso, R. B., Buana, I. K. S., & Pramadi, Y. R. (2020). Android Malware Detection Using Hybrid-Based Analysis Deep Neural Network. *2020 3rd International Conference on Information and Communications Technology, ICOIACT 2020*, 252–256. https://doi.org/10.1109/ICOIACT50329.2020.9332066

Han, H., Park, S., & Park, M. (2020). *Enhanced Android*

Malware Detection : An SVM-based Machine Learning Approach. *December 2016*, 75–81. https://doi.org/10.1109/BigComp48618.2020.00-96

Han, Q., Subrahmanian, V. S., & Xiong, Y. (2020). *Android Malware Detection via (Somewhat) Robust Irreversible Feature Transformations*. *6013*(iii), 1–16. https://doi.org/10.1109/TIFS.2020.2975932

Haq, I. U. L., Khan, T. A., Akhunzada, A., & Member, S. (2021). *A Dynamic Robust DL-based Model for Android Malware Detection*. *4*. https://doi.org/10.1109/ACCESS.2021.307937

Hashem, A., & Fiky, E. (2021). *Detection o f Android Malware using Machine Learning*. 9–16. https://doi.org/10.1109/MIUCC52538.2021.9447661

İbrahim, M., Issa, B., & Jasser, M. B. (2022). *A Method for Automatic Android Malware Detection Based on Static Analysis and Deep Learning*. *10*(October). https://doi.org/10.1109/ACCESS.2022.3219047

Iqbal, S., & Zulkernine, M. (2019). SpyDroid: A Framework for Employing Multiple Real-Time Malware Detectors on Android. *MALWARE 2018 - Proceedings of the 2018 13th International Conference on Malicious and Unwanted Software*, 33–40. https://doi.org/10.1109/MALWARE.2018.8659365

Jerbi, M., Chelly, Z., Bechikh, S., & Ben, L. (2020). Computers & Security On the use of artificial malicious patterns for android malware detection. *Computers & Security*, *92*, 101743. https://doi.org/10.1016/j.cose.2020.101743

K, S. J. (2020). *based Android Malware Detection*. *Icoei*. https://doi.org/10.1109@ICOEI48184.2020.9142929

Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering - A systematic literature review. *Information and Software Technology*, *51*(1), 7–15. https://doi.org/10.1016/j.infsof.2008.09.009

Kumar, S., & Thomas, C. (2021). ProDroid — An Android malware detection framework based on profile hidden Markov model. *Pervasive and Mobile Computing*, *72*, 101336. https://doi.org/10.1016/j.pmcj.2021.101336

Mobile Anti-Malware Market - Global Industry Analysis, Size, Share, Growth, Trends, and Forecast, 2022 - 2028 - MarketWatch. (2022.). Retrieved December 2, 2022, from https://www.marketwatch.com/press-release/mobile-anti-malware-market---global-industry-analysis-size-share-growth-trends-and-forecast-2022---2028-2022-11-21

Ndatsu, Z. & Adebayo, O. . (2020.). *Framework for the Detection of Android Malware Using Artificial Immune System*. *2017*, 117–126.

Olawale Surajudeen, A. (2012). Malware Detection, Supportive Software Agents and Its Classification Schemes. *International Journal of Network Security & Its Applications*, *4*(6), 33–49. https://doi.org/10.5121/ijnsa.2012.4603

Page, M. J., McKenzie, J. E., Bossuyt, P., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., … Moher, D. (2021). The prisma 2020 statement: An updated guideline for reporting systematic reviews. *Medicina Fluminensis*, *57*(4), 444–465. https://doi.org/10.21860/medflum2021_264903

Parker, C., McDonald, J. T., Johnsten, T., & Benton, R. G. (2019). Android Malware Detection Using Step-Size Based Multi-layered Vector Space Models. *MALWARE 2018 - Proceedings of the 2018 13th International Conference on Malicious and Unwanted Software*, 49–58. https://doi.org/10.1109/MALWARE.2018.8659372

Raghav, U., Martinez-Marroquin, E., & Ma, W. (2021). Static Analysis for Android Malware detection with Document Vectors. *IEEE International Conference on Data Mining Workshops, ICDMW*, *2021-Decem*, 805–812. https://doi.org/10.1109/ICDMW53433.2021.00104

Rathore, H. (2021). *Towards Robust Android Malware Detection Models using Adversarial Learning*. 424–425. https://doi.org/10.1109/PERCOMWORKSHOPS51409.2021.9430980

Ren, Z., Wu, H., Ning, Q., Hussain, I., & Chen, B. (2020). Ad Hoc Networks End-to-end malware detection for android IoT devices using deep learning. *Ad Hoc Networks*, *101*, 102098. https://doi.org/10.1016/j.adhoc.2020.102098

Ri, H., Rpx, E. L. O., Wu, H. G. X., Ri, H., Vdklq, G., Rpx, E. L. O., Wu, H. G. X., Ri, H., Dnoh, V., Rpx, E. L. O., Wu, H. G. X., Ri, H., Nlolf, H., Rpx, E. L. O., Wu, H. G. X., Rpxudo, P., & Frp, U. (2021). *Apk2Img4AndMal : Android Malware Detection Framework Based on Convolutional Neural Network*. 22–25. https://doi.org/10.1109/UBMK52708.2021.9558983

Şahİn, D. Ö., Akleylek, S., & Kiliç, E. (2022). *LinRegDroid : Detection of Android Malware Using Multiple Linear Regression Models-Based Classifiers*. *10*. https://doi.org/10.1109/ACCESS.2022.3146363

Sandeep, H. R. (2019). Static analysis of android malware detection using deep learning. *2019 International Conference on Intelligent Computing and Control Systems, ICCS 2019, Iciccs*, 841–845. https://doi.org/10.1109/ICCS45141.2019.9065765

Sharma, R. M., & Agrawal, C. P. (2022). A BPSO and Deep Learning Based Hybrid Approach for Android Feature Selection and Malware Detection. *Proceedings - 2022 IEEE 11th International Conference on Communication Systems and Network Technologies, CSNT 2022*, 628–634. https://doi.org/10.1109/CSNT54456.2022.9787671

Shatnawi, A. S., Jaradat, A., Yaseen, T. B., Taqieddin, E., Al-ayyoub, M., & Mustafa, D. (2022). *An Android Malware Detection Leveraging Machine Learning*. *2022*. https://doi.org/https://doi.org/10.1155/2022/1830201

Shyong, Y., Jeng, T., & Chen, Y. (2020). *Combining Static Permissions and Dynamic Packet Analysis to Improve Android Malware Detection*. 75–81.

Sihag, V., Justice, C., Vardhan, M., Singh, P., & Choudhary, G. (2021). *De-LADY : Deep learning based Android malware detection using Dynamic De-LADY : Deep learning based Android malware detection using Dynamic features*. *May*. https://doi.org/10.22667/JISIS.2021.05.31.034

Song, Q. (2021). *DroidRadar : Android Malware Detection Based on Global Sensitive Graph Embedding*. 802–809. https://doi.org/10.1109/TrustCom53373.2021.00115

Srinivasan, R., Karpagam, S., Kavitha, M., & Kavitha, R. (2022). An Analysis of Machine Learning-Based Android Malware Detection Approaches. *Journal of Physics: Conference Series*, *2325*(1). https://doi.org/10.1088/1742-6596/2325/1/012058

Srivastava, R., Mishra, R. P., Kumar, V., Shukla, H. K., Goyal, N., & Singh, C. (2020). Android malware detection amid COVID-19. *Proceedings of the 2020 9th International Conference on System Modeling and Advancement in Research Trends, SMART 2020*, 74–78. https://doi.org/10.1109/SMART50582.2020.9337105

Taheri, R., Ghahramani, M., Javidan, R., & Shojafar, M. (2020). Similarity-based Android malware detection using Hamming distance of static binary features. *Future Generation Computer Systems*, *105*, 230–247. https://doi.org/10.1016/j.future.2019.11.034

Wang, S., Chen, Z., Yan, Q., Ji, K., Peng, L., & Yang, B. (2020). *Deep and broad URL feature mining for android malware detection*. *513*, 600–613. https://doi.org/10.1016/j.ins.2019.11.008

Wang, W., Zhao, M., & Wang, J. (2018). Effective android malware detection with a hybrid model based on deep autoencoder and convolutional neural network. *Journal of Ambient Intelligence and Humanized Computing*, *0*(0), 0. https://doi.org/10.1007/s12652-018-0803-6

Wang, X., & Li, C. (2021). Neurocomputing Android malware detection through machine learning on kernel task structures. *Neurocomputing*, *435*, 126–150. https://doi.org/10.1016/j.neucom.2020.12.088

Xu, J., Li, Y., Deng, R. H., & Xu, K. (2022). SDAC: A Slow-Aging Solution for Android Malware Detection Using Semantic Distance Based API Clustering. *IEEE Transactions on Dependable and Secure Computing*, *19*(2), 1149–1163. https://doi.org/10.1109/TDSC.2020.3005088

Xu, Z., Li, M., Hei, Y., Li, P., & Liu, J. (2021). *A Malicious Android Malware Detection System based on Implicit Relationship Mining*. 59–64. https://doi.org/10.1109/CSCloud-EdgeCom52276.2021.00021

Yuan, W., Jiang, Y., Li, H., & Cai, M. (2021). A Lightweight On-Device Detection Method for Android Malware. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *51*(9), 5600–5611. https://doi.org/10.1109/TSMC.2019.2958382

Zhang, H., Li, S., Wu, X., Han, W., & Wang, L. (2021). An android malware detection approach using multi-feature fusion and TF-IDF algorithm. *Proceedings - 2021 IEEE 6th International Conference on Data Science in Cyberspace, DSC 2021*, 629–634. https://doi.org/10.1109/DSC53577.2021.00100

Zhang, J., Zhang, C., Liu, X., Wang, Y., Diao, W., & Guo, S. (2021). *S HADOW D ROID : Practical Black-box Attack against ML-based Android Malware Detection*.

Zhang, N., Tan, Y. an, Yang, C., & Li, Y. (2021). Deep learning feature exploration for Android malware detection. *Applied Soft Computing*, *102*, 107069. https://doi.org/10.1016/j.asoc.2020.107069

Zhang, Y., Sui, Y., Pan, S., Zheng, Z., Ning, B., Tsang, I., & Zhou, W. (2020). Familial Clustering for Weakly-Labeled Android Malware Using Hybrid Representation Learning. *IEEE Transactions on Information Forensics and Security*, *15*(XXX), 3401–3414. https://doi.org/10.1109/TIFS.2019.2947861

Zhou, H., Yang, X., Pan, H., & Guo, W. (2020). *An Android Malware Detection Approach Based on SIMGRU*. 0–6. https://doi.org/10.1109/ACCESS.2020.3007571

Zhou, J., Niu, W., Zhang, X., Peng, Y., Wu, H., & Hu, T. (2020). Android Malware Classification Approach Based on Host-Level Encrypted Traffic Shaping. *2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing, ICCWAMTIP 2020*, 246–249. https://doi.org/10.1109/ICCWAMTIP51612.2020.9317429

Zhu, H., Li, Y., Li, R., Li, J., You, Z., & Song, H. (2021). SEDMDroid: An Enhanced Stacking Ensemble Framework for Android Malware Detection. *IEEE Transactions on Network Science and Engineering*, *8*(2), 984–994. https://doi.org/10.1109/TNSE.2020.2996379

Zhu, H., Wang, L., Zhong, S., Li, Y., & Sheng, V. S. (2022). *for Android Malware Detection*. *34*(12), 55585570https://doi.org/10.1109/TKDE.2021.3067658