INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

REVIEW ARTICLE

A Vast Review of Recognizing the Presence of Android Malware Based on Ensemble Machine Learning Technique

Saqib Malik[1]*, **Narendra Sharma**[2]

1 Research Scholar, Department of Computer Application, Sri Satya Sai University of Technology and Medical Sciences, Sehore, Madhya Pradesh, India
2 Associate Professor, Department of Computer Science, Sri Satya Sai University of Technology and Medical Sciences, Sehore, Madhya Pradesh, India

OPEN ACCESS

Received: 21-09-2023
Accepted: 14-12-2023
Published: 12-01-2024

Citation: Malik S, Sharma N (2024) A Vast Review of Recognizing the Presence of Android Malware Based on Ensemble Machine Learning Technique. Indian Journal of Science and Technology 17(2): 149-165. https://doi.org/10.17485/IJST/v17i2.2406

* **Corresponding author**.

sayyed.saqib531@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Background: It is evaluated that there is 70% to 80% of smartphone users have an Android mobile. Given its trend, a lot of malware strikes on the Android OS. In 2018, the largest number of malware attacks was identified, when there were 10.5 billion such malicious activity detected worldwide. Machine learning has emerged as a promising approach for detecting Android malware, and Ensemble machine learning has been shown to enhance the accuracy of malware detection in other domains. **Objectives:** In this paper, the systematic literature review were conducted using natural language processing. 30 papers are selected from January 2019 to August 2023 to give a clear picture of the most recent work in Android malware detection using ensemble machine learning. **Methods:** Initially the ensemble machine learning analysis were categorized in Android malware detection into four groups. Static Ensembles, Dynamic Ensembles, Hybrid Ensembles and Structural Ensembles method and compare the outcomes of empirical evidence with the help of a systematic literature review using the natural language processing method. **Findings:** The findings demonstrate an emerging trend of using NLP for Android malware detection in combination with ensemble machine learning models. The use of natural language processing (NLP) enhances the capacity to identify harmful patterns by making it easier to extract key features from textual input. The paper also emphasizes the variety of ensemble models used, including Tree-Based, Meta ensemble, Specialized ensemble and others. **Significance :** The novel aspects of this paper are its extensive comparative evaluation of ensemble and non-ensemble models, its original combination of NLP and ensemble machine learning for Android malware detection, and its extensive review of the literature with an eye toward future directions and research gaps. As a result, based on the present research community, it is important to develop some unique ways to enhance Android malware detection.

Keywords: Ensemble Machine Learning; Static analysis ensemble; Dynamic Analysis Ensembles; Hybrid Feature Ensembles; Structural Analysis Ensembles

1 Introduction

Smartphone users have expanded rapidly in previous years. Subsequently, the ubiquity of aggressors expanded as clients expanded. Before August 2023, an expected 78.50% of advanced cell phone clients used the Android working framework. Machine learning has arisen as a promising methodology for identifying Android malware, as it can figure out how to perceive designs in enormous datasets of uses. Ensemble machine learning is a procedure that joins different classification algorithms to work on the improvement of accuracy and has been demonstrated to be successful for malware recognition in different spaces. The area of malware detection is an unending challenge among aggressors and against malware designers. That's what to recall "In the year 2023 there are 2700 most natural Android malware dangers are happen" when AV-Comparatives utilized the datasets. Creator likewise sees that benchmark handily accomplished when the identification rate lies somewhere in the range of 90% to 100% [https://www.av-comparatives.org/tests/mobile-security-review-2023/].

Threat Report of the Zimperium 2023 Global Mobile [https://get.zimperium.com/ 2023-global-mobile-threat-report/.] evaluates the third-party market data, the other party insights, and footage from top professionals in the field in addition to research from Zimperium's z Labs team. It also looks at trends that have changed the smartphone security environment during the past decade. Last year, Zimperium discovered 2,000 samples of malware weekly that weren't currently recognized by the industry.

1.1 Research Gaps

To reflect on the detection of Android malware using ensemble machine learning, initially, various surveys and evaluations were studied that were written about Android malware detection with the ensemble approach. Chowdhury et al.[(1)] give a comparison of the effectiveness of the different malware identification strategies and analyze the performance assessment criteria to evaluate the usefulness associated with their methods. They also give a summary of the present state of malware detection for Android using machine learning techniques. By integrating Support Vector Machine and Random Forest in the process of detection, Mijoya et al.[(2)] suggest a model for detecting malware using high accuracy and fewer false positives. They also give a review of methods of machine learning in malware detection in Android devices. The literature on machine learning-based malware detection in smart manufacturing was thoroughly reviewed by Sangeeta Rani et al.[(3)]. They located and examined 117 related papers that were presented at conferences and in scholarly publications. Subsequently, they also examine the various machine learning (ML) approaches for detecting malware, encompassing static, dynamic, and hybrid analytic methodologies. They also talk about the difficulties and potential developments in machine learning-based malware detection for smart industrial settings. Ali Muzaffar et al.[(4)] examine the prior studies that adopted machine learning to identify Android malware in their study. They categorize them based on how they employ static, dynamic, or hybrid characteristics and provide an overview of supervised unsupervised, deep learning, and online learning methodologies. Such as Ayat Droos et al.[(5)] suggest a novel approach for identifying Android malware through machine learning classifiers. They follow a strategy to categorize each APK program as harmful or genuine, and it demonstrates that the Random Forest method offers the highest accuracy. Al-garadi et al.[(6)] reviewed the use of natural language processing (NLP) in malicious domain identification. They discovered that while NLP techniques have the ability to be a useful tool for this endeavor, there are a number of issues that still need to be resolved. These challenges involve the dynamic nature of malicious domains, the dearth of extensive labeled data, and the computational expense of natural language processing techniques. Sen and Can[(7)] also examine how to improve Android security by utilizing natural language processing (NLP) approaches. They contend that NLP may be used to address a variety of security issues like Android malware detection. They addressed several challenges like Evolving Nature of Threats, Data Availability and Computational Complexity.

However, the study conducted on Android malware detection in the past few years has a gap. Consequently, it is critical to describe Android malware detection with ensemble machine learning techniques over the past decade.

1.2 Research Questions

Research questions identify the topics that must be examined in this review, and the responses to these questions serve as the foundation for the discussion to be examined in this review, and the responses from these questions serve as an outline for the discussion part.Table 1 lists five research questions about the use of ensemble machine-learning techniques to find malware for Android devices.

Table 1. Research Questions

Q. No	Research Questions	Direction

Continued on next page

Table 1 continued

Q1.	What are the most widely used Ensemble analysis Methods in Android malware detection?	Determine the type of Ensemble analysis Methods used in Android malware identification.
Q2.	How are the empirical evaluations of Android malware detection utilizing Ensemble analysis carried out in these primary studies?	Establishes the overall methodology for evaluating the empirical data in the original investigations.
Q2.1	What type of datasets are utilized to identify Android malware?	Specify the proper experimental datasets.
Q2.2	Which supporting tools are utilized for Android malware detection using ensemble analysis?	Analyze the proper support tools for ensemble analysis.
Q2.3.	What features are frequently applied to identify Android malware?	Analyze the frequently applied features.
Q2.4	What types of models are utilized for the identification of Android malware?	Analyze the frequently utilized models.
Q2.5	Which performance metrics are employed in the Ensemble analysis-based Android malware identification process?	Analyze the performance metrics.
Q3.	How effective are ensemble analysis approaches for detecting Android malware according to empirical evidence?	Evaluate the effectiveness of ensemble machine learning methods to detect Android malware is evaluated using conventional performance metrics.
Q4.	wherever Model I outperform Model II in Android malware detection according to empirical evidence?	Seeks to determine if Model I outperform Model II using a similar dataset and performance metric.

1.3 Research Contributions

Following the identification of closely related works, the systematic literature review with natural language processing were conducted to gain a comprehensive understanding of Android malware detection using ensemble methods during the last several years. The following is a list of this review's key contributions.

- The fundamental components of Android malware detection via the ensemble method are the basis for this literature review and talk about the weaknesses and strengths of this methodology.
- According to the features of the apps, this paper classifies ensemble machine-learning techniques in Android malware into four groups. After that, this study evaluates ensemble analysis capacity to identify malware and contrasts the effectiveness of several approaches for Android malware detection by examining the empirical data.
- The results of empirical data show that the ensemble machine learning method which joins different classifiers to make more exact expectations is effective in detecting new malware rather than Static or Dynamic analysis individually.
- In conclusion, utilizing the ensemble machine learning method, the future work was analyzed and outlined in the detection of Android malware and ensemble learning can successfully improve the location precision and strength of malware discovery frameworks.

The survey paper is organized into 5 fundamental segments, incorporating the first section with the introduction. The second section portrays the research question and comprehensive literature review method. The third area is devoted to results. The fourth area to discussion and challenges and the last area is to the conclusion and remarks.

2 Review Method

This review is based on the Systematic Literature Review with natural language processing method. The entire SLR-NLP procedure, which is broken down into three steps, is depicted inFigure 1.

- Preparation of the review. This step tries to establish the Strategy and identify the objective for this review.
- Performing the review. This section highlights the main focus findings in this review, which can be broken down into six phases.

1. Identify research questions. Research questions identify the topics that must be examined in this review, and the responses to these questions serve as the foundation for the discussion to be examined in this review, and the responses from these questions serve as an outline for the discussion part.
2. Develop a search strategy. To gather primary research, this step identifies search media and search phrases.

3. Screen and select studies. Both inclusion and exclusion criteria are used in this selection of studies. Irrelevant papers were filter under the inclusion and exclusion criteria to decide which studies are included or excluded throughout this review.
4. Extract data from studies using NLP. This step's goal is to create a data extraction form that would correctly capture the data associated with the study questions. Data from studies may be extracted from several file types, such as PDF files, text files, and HTML files, using NLP methods.
5. Analyze data with NLP. After extracting the data from studies, the data ware analyzed to answer our research questions. This can be done using NLP techniques.
6. Synthesize findings. The objective of this phase is to compile and summarize the findings of the main research.

   (a) Review Reporting. This phase's goal, based on these criteria, is to fulfill this review.

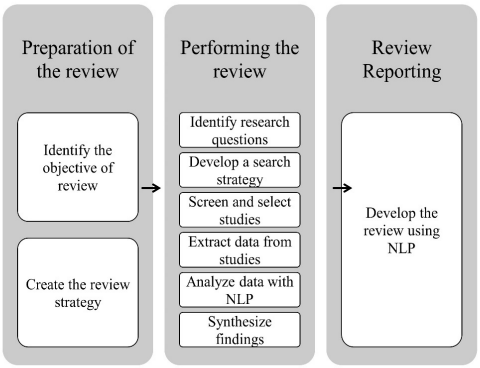


Fig 1. Outline of this review

2.1 Identify Research Questions

The major purpose of this portion is to develop the research questions that will be utilized to select the main investigations and make up the bulk of the report. Table 1 lists five research questions about the use of ensemble machine-learning techniques to find malware for Android devices. Q1 defines the type of ensemble learning technique used in the identification of malicious software for Android. Q2 establishes the overall methodology for evaluating the empirical data in the original investigations. There are six aspects to this subject, each of which focuses on a different topic, such as experimental datasets, support tools for ensemble machine learning, frequently utilized features, reduction of features approaches, chosen models to identify malware for Android, and overall performance metrics. We investigate the Q3 and Q4 based on empirical data. Regarding Q3, the effectiveness of ensemble machine-learning methods to detect Android malware is evaluated using conventional performance metrics. Q4 seeks to determine if Model I outperform Model II using a similar dataset and performance metric.

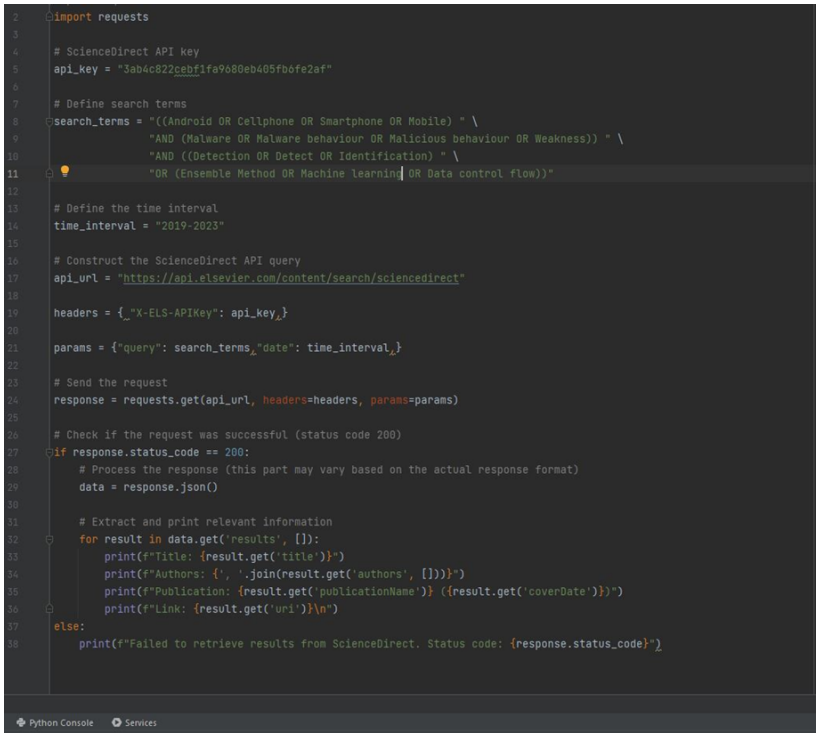
2.2 Develop A Search Strategy

We created some search keywords linked to this study to find papers on Android malware detection using the ensemble machine learning method. The primary strategy is to use Natural Language Processing to apply Boolean expressions—including ”AND” and ”OR”— to search term combinations. The search phrases (Android OR Cellphone OR Smartphone OR Mobile) AND (Malware OR Malware behaviour OR Malicious behaviour OR Weakness) AND (Detection OR Detect OR Identification) AND (Ensemble Method OR Machine learning OR Data control flow) may be distilled into a few basic categories. The appropriate digital archives are chosen after making sure such search terms. The following is a list of the five online databases that we searched. We use an API key and URL of the above mention databases and the python programming language as the interface.

- Web of Science
- Scopus
- SpringerLink

- Google Scholar
- ScienceDirect
- IEEE Xplore Digital Library

The search is conducted on the five online databases listed earlier, including major publications and conferences. Such publications and conferences are primarily from Computer security and Computer Science and Engineering and Mobile Security and Machine learning. In the reference area, in which studies make up a minor percentage of the major research, we also compile the studies that are connected to malware detection for Android using ensemble machine learning. The search period is from January 2019 to August 2023, and all research about search terms are included. Figure 2 shows the implementation of these Boolean expressions through NLP. We search each database individually so Figure 2 shows only Science direct database. We also search from other databases like IEEE and web of science using its API with same search terms.



```python
import requests

# ScienceDirect API key
api_key = "3ab4c822cebf1fa9680eb405fb0fe2af"

# Define search terms
search_terms = "((Android OR Cellphone OR Smartphone OR Mobile) " \
               "AND (Malware OR Malware behaviour OR Malicious behaviour OR Weakness)) " \
               "AND ((Detection OR Detect OR Identification) " \
               "OR (Ensemble Method OR Machine learning OR Data control flow))"

# Define the time interval
time_interval = "2019-2023"

# Construct the ScienceDirect API query
api_url = "https://api.elsevier.com/content/search/sciencedirect"

headers = {"X-ELS-APIKey": api_key}

params = {"query": search_terms, "date": time_interval}

# Send the request
response = requests.get(api_url, headers=headers, params=params)

# Check if the request was successful (status code 200)
if response.status_code == 200:
    # Process the response (this part may vary based on the actual response format)
    data = response.json()

    # Extract and print relevant information
    for result in data.get('results', []):
        print(f"Title: {result.get('title')}")
        print(f"Authors: {', '.join(result.get('authors', []))}")
        print(f"Publication: {result.get('publicationName')} ({result.get('coverDate')})")
        print(f"Link: {result.get('uri')}\n")
else:
    print(f"Failed to retrieve results from ScienceDirect. Status code: {response.status_code}")
```


Fig 2. Implementation of the Boolean expressions

2.3 Screen and Select Studies

We create three criteria for inclusion and base our search on five online databases to choose the linked papers. First, the title, abstract, and keywords all include search terms. Second, the research introduces ensemble machine-learning tools. Third, the studies include conducting empirical experiments. All the Inclusion criteria are shown in Table 2 along with their eligibility. We also figure out three exclusion criteria that are unrelated to the goal of this review. First, the research that has not been published in English is filtered out. Second, we do not include shorter research with duplicated articles. In general, some research is released simultaneously in journals and conferences. We identify this research based on authors, titles, and abstracts, with less thorough studies being left out. Third, Certain studies for the operating system known as Android are excluded as a result of the word ”Android” within the search keywords. Also, the Exclusion criteria are shown inTable 3 along with their eligibility.

Table 2. Inclusion Criteria

Criterion	Eligibility	Inclusion
Literature type	Peer-reviewed journals, conference proceedings and technical reports.	Studies published in reputable peer-reviewed journals, conferences, proceedings and technical reports that meet the inclusion criteria.

Continued on next page

Table 2 continued

Language	English	Studies available in English.
Timeline	January 2019 to August 2023	Studies published within the specified timeline.
Solution Oriented	Studies must meet three criteria: 1. Title, abstract, and keywords include search terms. 2. The research introduces ensemble machine-learning tools. 3. The studies include conducting empirical experiments.	Studies that satisfy the three specified criteria: 1. Title, abstract, and keywords must include the specified search terms. 2. The research must introduce ensemble machine-learning tools. 3. The studies must include conducting empirical experiments.

Table 3. Exclusion Criteria

Criterion	Eligibility	Exclusion
Literature type	Peer-reviewed journals, conference proceedings and technical reports.	Shorter research with duplicated articles is excluded. Studies released simultaneously in journals and conferences, identified based on authors, titles, and abstracts, with less thorough studies being left out are excluded. Certain studies for the Android operating system are excluded due to the word "Android" within the search keywords.
Language	English	Studies not published in English are excluded.
Timeline	January 2019 to August 2023	Studies outside the specified timeline are excluded.
Solution Oriented	Studies must meet three criteria: 4. Title, abstract, and keywords include search terms. 5. The research introduces ensemble machine-learning tools. 6. The studies include conducting empirical experiments.	Studies that do not meet any of the three specified criteria are excluded.

2.4 Extract Data from Studies using NLP

Designing forms for correctly obtaining the data from the initial study refers to the data extraction procedure. This section discovered solutions to research questions based on the data in the data extraction forms.

- The studies' Author, publication date, and publishing source. In this section, we use NLP techniques, such as Named Entity Recognition (NER) to find out who wrote the article, when it was published, and where it was published—including, for primary research, journals and conferences.
- Ensemble machine learning techniques. This section primarily focuses on the Ensemble machine learning methods used in initial research. We group Ensemble machine learning methods based on the app's features.
- Empirical evidence. The data was synthesized from six sources to address these research questions: Experimental datasets, Ensemble learning support tools, feature elimination strategies, Ensemble analysis features, utilized models, and performance metrics.

The findings of the data extraction and synthesis are then stored in a worksheet. At last, to display the decisive results, we employ the pie chart and table.

3 Results and Discussion

3.1 Results

The purpose of this part is to show the findings from the preliminary research. We begin by describing the primary research. Factually speaking, we next provide this review outcome and findings by giving the answer of the research's questions.

3.1.1 Overview of Studies

45 studies are found on ensemble machine learning to detect malware. 25 studies have been discussed in this section in terms of publishing source and year of publication.

3.1.1.1 Publication Year. Figure 3 **(a)** demonstrates that there will be 7,8,9,10 and 11 studies from 2019 to 2023, accordingly. This figure shows that the number of studies in 2023 contributes to the biggest share. The number of research linked to Android malware detection using Ensemble analysis has increased significantly from 2019 to 2023, from the above time interval, there has been a significant increase in the amount of research evaluating ensemble analysis's effectiveness in detecting Android malware. This data indicates that throughout the past few years, Android malware detection has consistently been a popular issue.

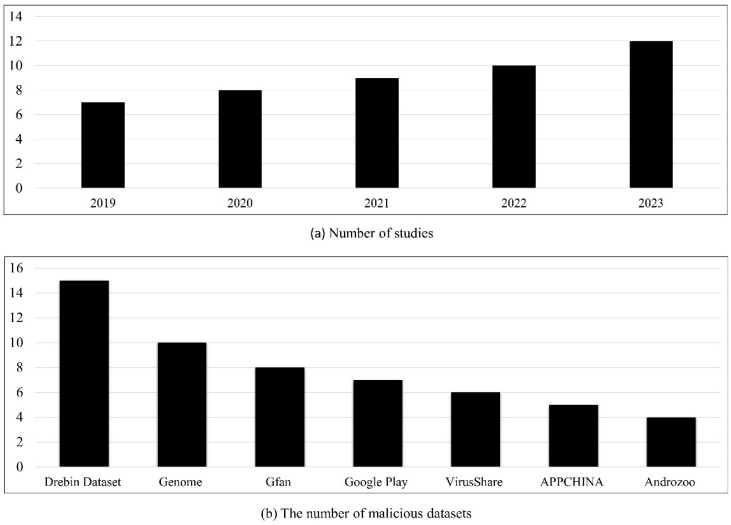


Fig 3. (a)Number of studies, (b) The number of malicious datasets in the primary studies

3.1.1.2 Source of Publication. Table 4 summarizes the key publishing type of source including the total quantity of studies that are connected to it in the primary studies. This table shows that the top 5 sources of publications are IEEE Transactions on Information Forensics, Computer and Security, Wireless Personal Communications, and IEEE Access Security and Privacy, with a total of 9, 7, 6, 5 and 6 respectively. Many additional publication sources, including the IEEE International Conference on Communications (ICC) and the International Conference on Wireless Networks (ICWN), are not included, as is to be expected. The number of studies from the journal represents over 50% of the total, based on the data in this table.

Table 4. The outcomes of the source of publications

Amount Published	Publishers	Journals	Year Range
7	Web of Science	IEEE Transactions on Information Forensics and Security, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Mobile Computing, IEEE Access.	2019-2023
9	Scopus	IEEE Transactions on Information Forensics and Security, IEEE Transactions on Dependable and Secure Computing, IEEE Access, ACM Transactions on Information and System Security	2019-2023
6	SpringerLink	Journal of Information Security and Applications, Security and Communication Networks, Peer-to-Peer Networking and Applications, Journal of Network and Computer Applications	2019-2023
5	ScienceDirect	Computers & Security, Journal of Network and Computer Applications, Computers & Mathematics with Applications, Information Sciences	2019-2023

Continued on next page

Table 4 continued

6	IEEE Xplore Digital Library	IEEE Transactions on Information Forensics and Security, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Mobile Computing, IEEE Access	2019-2023
1	Tsinghua University Press	Tsinghua Science and Technology	2019-2023
1	Wiley	Security and Communication Networks	2019-2023
2	ACM Conference	ACM Conference on Computer and Communications Security (CCS)	2019-2023
2	Conference	IEEE Symposium on Security and Privacy (S&P)	2019-2023
1	Conference	Network and Distributed System Security (NDSS)	2019-2023
1	Conference	ACM Conference on Data and Application Security and Privacy (CODASPY)	2019-2023
1	Conference	International Conference on Security and Privacy in Communication Systems (SecureComm)	2019-2023
1	Conference	International Conference on Security and Privacy in Communication Systems (SecureComm)	2019-2023
1	Conference	European Symposium on Research in Computer Security (ESORICS)	2019-2023
1	ACM Conference	ACM Conference on Mobile Computing and Networking (MobiCom)	2019-2023

3.1.2 Q1: What are the most widely used Ensemble Analysis Methods in Android malware detection?

Ensemble machine learning is an approach that combines numerous separate machine learning models to produce a more robust and accurate prediction model. The core concept underlying ensemble learning is to use the variety and combined knowledge of several models to enhance prediction accuracy. These models, which are also known as ”base classifiers” or ”weak learners,” are trained on the same dataset or different dataset subsets. To reach a judgment, the ensemble technique then integrates all of the forecasts, which is often more accurate and trustworthy than the predictions from the separate models. The ensemble analysis method was categorized into four groups based on features extracted from APKs, that involve Static Analysis Ensembles, Dynamic Analysis Ensembles, Hybrid Feature Ensembles and Structural Analysis Ensembles method.

A. Static Analysis Ensemble Method

Static analysis is a method that is done by examining the task without running the program. It surrounds a comprehensive area of methods that explore the dynamic nature of a program before its execution. As an implication, each static method must increase malware detection while decreasing the hazard of false positive. In prospects of 5 years, using a static analysis approach there were a lot of mechanisms have been built up to trace the complications of malware detection. These tools are divided into two broad categories.

A.1 Tools that depend predominantly on code study, like byte-code analysis.

A.2 API calls and permissions-based Tools.

We emphasize that this arrangement is a little crude, and works just to conduct the survey, instead of as a field of procedural perspective. Yet the evaluation of thought theory in the recent 5 years can be emphasized by our analysis.

A.1 Byte-Code Analysis-Based Methods

This is the first classification of tasks that focus upon the analysis of the app's code, also at the byte-code level or source-code level. The most typical works are discussed by several analyses. BFEDroid [8] is one of these proposed embedded static techniques of malware detection for mobile applications that depends on a 2-step process first is an abstraction of machine instructions, succeeded by a phase of machine learning. Yuxin Ding et al. [9] extract features from the bytecode picture using a temporal convolution network (TCN). Deep learning models that are suitable for processing linear data, like bytecode instructions, are called TCNs. They demonstrate that their approach is more effective than conventional Android malware detection techniques and can identify malware even if it is encrypted or obfuscated.

A.2 API calls and permissions-based method

This is a second type of static-based analysis method that covers the permissions analysis asked by the app and several calls of API that arise in its byte code. Shatnawi et al. [10] proposed an approach for static-based malware detection Android

permission and API calls. It inspects many semantic features of the application to discover and classify any malware, the list of API calls included by static features appearing in the code, the requested permission, and the app components set. Eslam Amer.[11] proposes an approach based on permissions to detect Android malware using an ensemble-based voting model. The author first gathers the combinations of the permission that both benign and malicious apps frequently ask for. Analyzing a big dataset of each harmless and malicious Android application allows for this. After their collection, the permission combinations are utilized to train an ensemble model. After that, these data are classified within the under-mentioned classification algorithm:

- Logistic Regression.
- K Nearest Neighbour.
- Support Vector Machine (SVM).
- Decision tree classifier.

Several more studies ([12–14]) have been selected and appear in Table 3 that use the static ensemble method.

B. Dynamic Ensemble Analysis Method

Dynamic analysis is implementing the problem in a running state. This type of task runs in a sandbox or controlled atmosphere. In many situations, dynamic analysis needs more resources and tools for computing than static analysis. Present Research work for anti-malware detection can additionally focus on static and dynamic analysis features otherwise associate both. Dynamic analysis is a revolutionary approach for malware detection, when we want to learn its nature and its ramifications on its environment, it requires running the program. It has some limitations like coverage limitations, such as it executes a single program at a time on behalf of every feasible program execution. We also organize dynamic tools in two large categories:

B.1 System Call-Based Methods

The first category is associated with the monitoring of system calls. There were two types of works detected in this category.

B.1.1 Call Logs for System

Using homogeneous and heterogeneous ensemble machine learning, Bhat et al.[14] suggested a system call-based Android malware detection technique. On a dataset of 10,000 malicious and 10,000 benign Android applications, their solution obtained an accuracy of 98.90%, which is greater than the accuracy of other cutting-edge malware identification approaches. Additionally, Pengfei Jing et al.[15] proposed a method of dynamic detection for Android terminal malware relying on the Native layer, in which the APK installing package of the application running on Android is analyzed, and the Native library and Dex files are fused and mapped into a 3-channel visual RGB colour image, and features of malware are extracted and fused to achieve dynamic malware detection.

B.2 System-level behaviours monitoring

System-level behaviour is a second phase where dynamic processes aim at information at the system level aside from system calls beneficial to identify harmful apps. In addition, many researchers involve system calls in their studies. Islam et al.[16] suggested a unique method of classifying Android malware that makes use of ensemble machine learning and optimal feature selection. Using a current dataset of malicious and benign Android applications, the authors tested their suggested method and obtained an accuracy of 94.48%, which is far greater than the modern facilities. The authors' method is resilient against code obfuscation methods, scalable, and applicable to big datasets of Android malware. A new method for detecting Android malware is presented by Zhou et al.[17], and it is based on the Summation of Multi-order Derivatives LSTM (SoMD-LSTM). The conventional LSTM model is improved by the SoMD-LSTM model. The app's execution activity is analyzed by the system to extract various aspects, such as system calls performed, network traffic created, and resources used. These characteristics show the app's dynamic behaviour. Maldy is another portable malware detection system that Karbab and Debbabi[18] introduced. It analyzes behavioural analysis reports using machine learning (ML) and natural language processing (NLP) techniques. Maldy uses natural language processing (NLP) techniques to extract characteristics from behavioural analysis reports and then uses machine learning algorithms to categorize the reports as harmful or benign.

Several more studies ([19,20]) have been selected and appear in Table 3 that use the Dynamic ensemble method.

C. Hybrid Ensemble Analysis

Hybrid analysis strategies for Android malware detection join various methods and ways to deal with work on the accuracy and effectiveness of recognizing malicious conduct in Android applications. Hybrid analysis combines static and dynamic analysis. In their study, Orieb AbuAlghanam et al.[21]proposed an ensemble learning-based malware detection system for Android. They use a hybrid analysis approach, combining both static and dynamic analysis techniques to extract features from Android apps. The two primary subsystems of the system—one for safe applications and the other for malicious apps—operate in parallel. The three classifiers that make up each subsystem's ensemble technique are OC-SVM, LOF, and modified isolation forest (M-iForest). In another study, Ramu Kuchipudi et al.[22] proposed a hybrid ensemble learning-based malware detection system for Android. In comparison to employing any one machine learning algorithm alone, the system achieves improved accuracy and better generalization by combining the strengths of numerous methods. The detection system's overall robustness

is enhanced and the influence of individual model biases is lessened. Zhu et al.[23] also proposed A multi-model ensemble learning framework (MEFDroid) for unbalanced Android malware detection. In this study, the drawbacks of traditional ensemble learning techniques are addressed by MEFDroid, which frequently prioritizes variety across base classifiers above accuracy. MEFDroid improves base classifier variety and accuracy at the same time, which boosts the overall performance of the malware detection system. In addition, Zhang et al.[24] present a hybrid sequence-based Android malware detection system that analyzes Android application code using machine learning and natural language processing (NLP). The system initially translates the Android application code into text sequences, from which it extracts features using natural language processing (NLP) techniques. Lastly, the system uses machine learning techniques to categorize code sequences as harmful or benign. Also, studies[25] and[26] have appeared in Table 3 that use the hybrid ensemble method.

D. Structural analysis Ensembles

A study in structural analysis ensembles highlights how important it is to comprehend how Android applications operate inside. These ensembles can spot unusual behaviours and patterns that can point to the existence of malware by using techniques including call graph analysis, control flow analysis, and data flow tracking. One important finding is that structural analysis can reveal malware's distinctive structural traits, including code obfuscation or hidden communication pathways. To identify privilege escalation, information leakage, and possibly dangerous inter-component relationships, different base classifiers might concentrate on different structural properties. Droid-MCFG is a malware detection for Android systems that uses manifest and control flow traces coupled with a multi-head temporal convolutional network (TCN) in the paper by Ullah et al.[27]. This method shows how to combine temporal characteristics with structural data for better identification of malware in Android applications. Mahindru et al.[28] created MLDroid, a thorough framework for Android malware detection that makes use of a variety of machine-learning approaches, as part of their research. This framework, which combines several machine-learning techniques and approaches, provides a reliable remedy for spotting Android malware. Nektaria Potha et al.[29] presented an extrinsic random-based ensemble technique for Android malware detection. They used an extrinsic random selection procedure to choose the top classifiers from a pool of base classifiers, the selection of extrinsic random procedure is a type of structural ensemble analysis and the method seeks to enhance the performance of conventional ensemble approaches. Given how often new malware families appear and how the malware environment is always changing, this method is very helpful for detecting Android spyware. Study[30] and[31] also appear in Table 5.

Table 5 lists the various Ensemble analysis methodologies together with the number and percent of studies that fall under each category. 30 studies are listed in this table. The following table demonstrates that the Static Analysis Ensemble method, which accounted for around 38% of all the research, is the most widely utilized ensemble analysis methodology. Additionally, there are around 27% and 20% of primary papers devoted to the dynamic analysis ensemble method and the Hybrid analysis ensemble method, respectively. The Structural Analysis Ensembles has received the fewest studies overall. It implies that the Static Ensemble method has received significant attention.

Table 5. The Category of Ensemble Analysis method

Category	Publications	No.	Percent
Static Analysis Ensembles	[8–13], [32–36]	11	38%
Dynamic Analysis Ensembles	[14–20], [37]	8	27%
Hybrid Feature Ensembles	[21–26]	6	20%
Structural Analysis Ensembles	[27–31]	5	13%

3.1.3 Q2: How are the empirical evaluations of Android malware detection utilizing Ensemble analysis carried out in these primary studies?

In answer to RQ2, we thoroughly review and evaluate the empirical data. The five phases of the method of empirical tests in the detection of malware on Android are:

- **Data collection:** Data collection's first goal is to compile both good and bad datasets. The outcomes are often more persuasive the more testing datasets available.
- **Feature extraction:** The second goal of feature extraction is to extract features from APKs by aiding ensemble analysis techniques.
- **Identify important characteristics:** Third, to identify and choose important characteristics, feature reduction techniques are used.
- **Model selection:** Fourth, model selection looks for a suitable model to discriminate between harmful and beneficial uses. These models include machine learning and statistical models.

- **Model assessment:** Fifth, model assessment seeks to judge how well models generalize based on performance metrics.

The next parts identify test datasets, tools for support in ensemble analysis, frequently used features, techniques for feature reduction, the utilized models, and performance measurements in the detection of Android malware through empirical studies.

3.1.4 Q2 1: What types of datasets are utilized to identify Android malware?

Researchers and scholars have used a variety of malware datasets in initial research studies. Datasets may be categorized into two groups based on their origin: in-lab datasets and in-the-wild datasets. In-lab datasets, which mostly comprise Drebin, Gfan and Genome, are widely recognized as baselines in Android malware detection. The first malware dataset to be compiled and made public is Genome. The first effective and understandable technique for detecting Android malware is Drebin, which relies on Genome Datasets found in the wild that are enhanced and organized constantly. The performance of techniques examined in the in-the-wild datasets is more trustworthy than that of the in-lab datasets because of the huge number of samples and various categories. Specific examples of datasets found in the wild include Virusshare, Google Play, Appchina and Androzoo. Furthermore, several researchers continue to make use of private datasets [9], [10]. When it comes to benign datasets, the majority of studies' apps come from the China Mobile Application Market and Google Play Store. Most benign datasets are not open source. This paper also designates whether the database consists of malware applications, Harmless applications or either malware and harmless. Upgrading the model often allows you to highlight the issue of concept drift, which occurs when a model used for prediction loses accuracy over time as the factors on which it relies become old and out of date. Figure 3 **(b)** displays the total quantity of datasets used in prior research. Some research repeatedly employs some malicious datasets to validate the accessibility and efficacy of suggested approaches. This statistic indicates that the most often used datasets in the lab are Drebin and Genome, whereas the most frequently used dataset in the wild is Virusshare.

3.1.5 Q2 2: Which supporting tools are utilized for Android malware detection using ensemble analysis?

In primary research, several tools are used to facilitate ensemble analysis. A few widely available tools can be used to carry out the primary ensemble analysis. These support tools are called Scikit-learn, MLFlow, NER, Natural Language Toolkit (NLTK), AndroBugs, MobSF, XGBoost, EMP library and Vote Ensemble. Additional processing and analysis are still required to acquire the Android application terminal representation format. It is found that Scikit-learn makes up the majority of the tools and is frequently used for basic NLP tasks, such as text vectorization, sentiment analysis, and text classification.

3.1.6 Q2 3: What features are frequently applied to identify Android malware?

APK behaviours may be represented by several feature groups at various levels. As a result, these characteristics' ability to identify malware is variable. Primary studies make use of many attributes to depict the behaviours of APKs to identify malware. As specified by four kinds of ensemble analysis methodologies in Q1, the feature selection procedure in a hybrid ensemble is intended to produce a thorough representation of the characteristics of an Android application, taking into account both static and dynamic factors. The ensemble learning framework then takes these aggregated characteristics as input. Additionally, Structural Analysis Ensembles is a challenging methodology that has been used in six studies; hence, this section concentrates on Android features and programs that employ hybrid-ensemble methods to find malware. More information on these aspects is provided in Figure 4 **(a), (b), (c)** and **(d).**

3.1.7 Q2 4: What types of models are utilized for the identification of Android malware?

There were many models used to detect malware but only ensemble models were analyzed for Android malware detection. The ensemble model based on machine learning has become more broadly applied to the detection of Android malware in the past decade due to the fast growth of these techniques in natural language processing, image detection, and other areas. These ensemble models categorize into four groups.

- Tree-based ensemble models.
- Meta-Ensemble Models.
- Specialized ensemble models.
- Other ensemble models.

Ensemble machine learning is a branch of machine learning that is extensively used in fields like image recognition and detection. It can also function effectively for the detection of Android malware. Regardless, resulting works have effectively accomplished high accuracy rates by using framework call sequences by Kim et al. [32] or composition of sequence and separate API calls and permission by Abubaker et al. [34]. These examinations utilized more suitable datasets to assess their system. In their study, Atacak et al. [26] also, connect static permissions and dynamic packet analysis and introduce an Android application

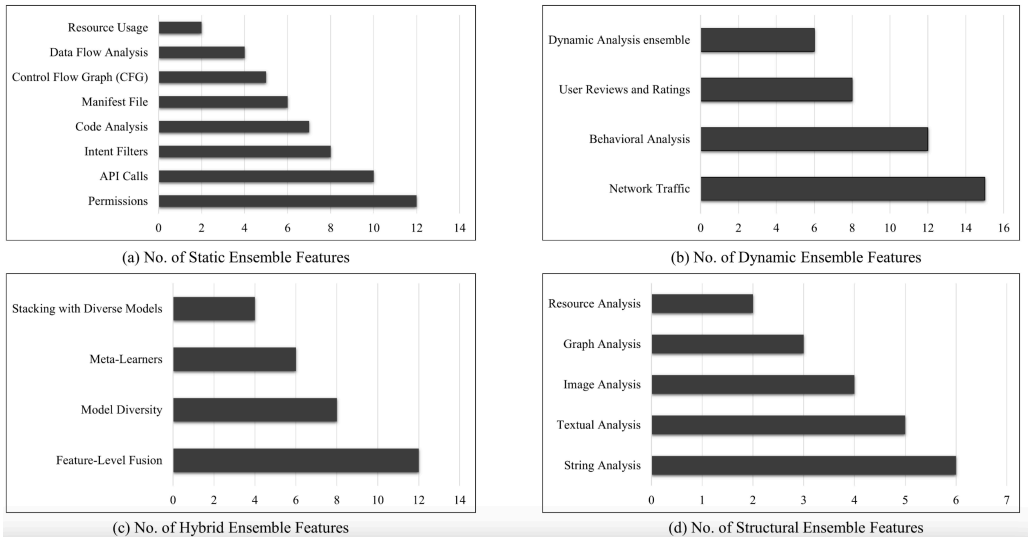


Fig 4. Features overview

classification system. In the first phase, they extracted the static permission features and made a permission vector for any application. This permission vector was then used to classify the application as either harmless or malware, utilizing a random forest model.

3.1.8 Q2 5: Which performance metrics are employed in the Ensemble analysis-based Android malware identification process?

The capacity for generalization of models for identifying malware on Android is assessed using performance metrics. The definition of performance measures and the number of research relevant to performance measures are presented below. We use the terms true positive (TP), true negative (TN), false positive (FP), and false negative (FN) to describe the definition of performance measures. TP stands for the number of harmless apps that were correctly identified as such, TN for those that were predicted falsely as malware, FP for those that were predicted falsely as benign applications, and FN for those that were correctly identified as such. Accuracy, Precision and recall are the most commonly used performance measures.

3.1.9 Q3: How effective are ensemble analysis approaches for detecting Android malware according to empirical evidence?

The purpose of this portion is to evaluate the efficacy of ensemble analysis methods for identifying malware on Android. The values of performance metrics are simply stored on the distinct dataset, since some studies have combined many datasets into a single dataset.

Then, accuracy, precision, and recall were selected as the assessment criteria and calculate the number of analyses that have used the same ensemble analysis approach. With a particular dataset, we solely store the values of performance metrics, which are then shown in distinct tables [Table 6]. Wherein Q2.5, are the three performance metrics most frequently employed. Using the same ensemble analysis method, finally determine the mean, standard deviation, minimum and maximum of three chosen performance metrics from the experiments. The number of investigations, as well as the mean, minimum, maximum, and standard deviation (Std.) of accuracy, precision, and recall, are shown in Table 6. In terms of mean values, the Static Analysis Ensembles approach performs the best scoring 97.57% for accuracy, 96.57% for precision, and 96.86% for recall. The efficacy of the dynamic analysis ensemble and hybrid analysis ensemble is comparable. Table 6 also demonstrates that the standard deviation in four ensemble analysis methodologies spans from 2.40 to 4.65, with an aggregate score above 88.30%. Ultimately, it indicates that methods for ensemble analysis are useful in identifying malware for Android devices.

Table 6. Performance metrics of all four categories of ensemble method

Type	Performance metrics	No. of studies	Mean	Min	Max	Std
Static Analysis Ensembles	Accuracy	7	98.67%	93.84%	99.81%	2.58

Continued on next page

Table 6 continued

	Precision	3	97.68%	93.42%	99.84%	2.66
	Recall	4	97.95%	91.38%	99.25%	3.44
	Accuracy	5	97.51%	88.62%	99.68%	4.78
Dynamic Analysis Ensembles	Precision	3	96.90%	87.75%	99.35%	4.99
	Recall	3	96.85%	89.56%	99.50%	4.20
	Accuracy	5	97.45%	91.38%	99.62%	3.48
Hybrid Feature Ensembles	Precision	3	96.42%	92.50%	98.89%	2.63
	Recall	2	96.84%	91.59%	98.86%	3.06
	Accuracy	3	94.93%	87.43%	98.25%	4.52
Structural Analysis Ensembles	Precision	2	93.95%	89.56%	98.54%	3.25
	Recall	3	95.50%	89.66%	97.62%	3.62

3.1.10 Q4: Where the Model I outperform Model II in Android malware detection according to empirical evidence?

To investigate this research issue the data is used from Q3. Through analysis and comparison, it is discovered that Model I is superior to Model II when the dataset's sample size is ignored. In addition, the models are separated into two groups: Ensemble machine learning models and non-Ensemble machine learning models. Next, using a similar dataset, we compute the averages of both of the models' accuracy, precision, and recall. Ultimately, we arrived at a tentative conclusion that the Ensemble machine learning models perform better than the non-Ensemble machine learning models after comparing the two models' performances. The overall performance is displayed in Figure 5 on Drebin, Androzoo and Virusshare datasets. In specifics, the Ensemble model performs better than the non-ensemble model, except for Google Play, Gfan, genome and Appchina. The Ensemble model's accuracy outperforms the non-Ensemble model, particularly on Drebin. While the Ensemble model outperforms the non-ensemble model in terms of accuracy on the Androzoo, the two models' accuracy is nearly equal. This chart shows that the Ensemble model performs better overall than the non-Ensemble model.

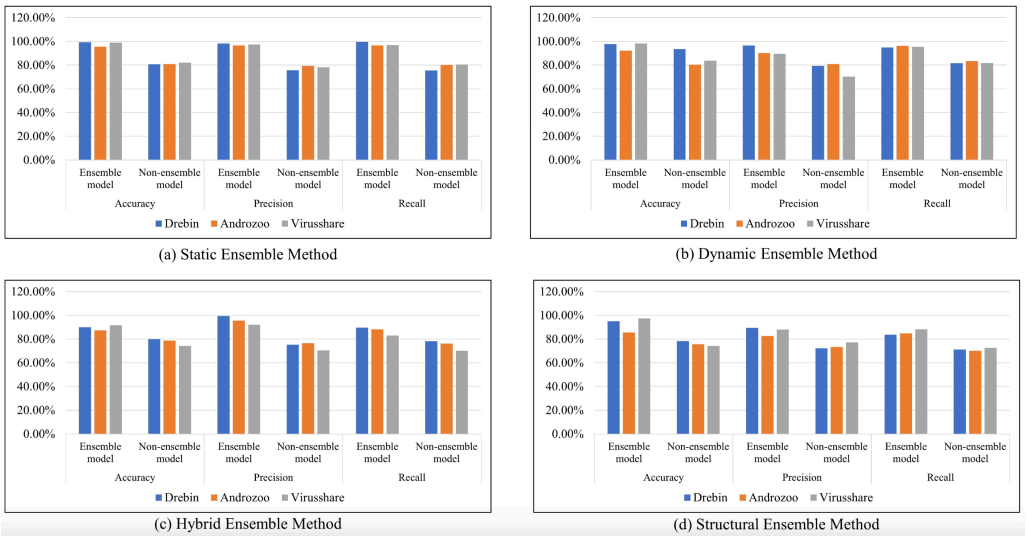


Fig 5. Overall performance of Static, Dynamic, Hybrid and Structural ensemble methods on Drebin, Androzoo and Virus share

3.2 Discussion

In the past few years, it has been widely discussed to use machine learning models to identify Android malware from Q2.4, as opposed to any other models. In particular, we note that the majority of studies in the field of machine learning models use non-ensemble models to identify malware. Applying a non-ensemble approach to Android malware detection is still unsatisfactory, though. This is noticed by Many researchers that the detection of malware with traditional techniques, like specification-based detection or signature-based detection, is insufficient to give efficient safety with the latest malware. Subsequently, Various methodologies and technologies that rely on behaviour-based analysis have been at the basis of malware detection in recent

years. In our opinion, ensemble analysis shows promise as a paradigm for Android detection of malware. It is advised to use ensemble analysis to identify Android malware to continually improve their robustness. This section includes many talks on the Performance, security, trade-offs, weakness, challenges and testing datasets of research topics aimed at improving Android malware detection using ensemble analysis.

Performance: According to our findings, Ensemble Models perform better than non-ensemble models regarding accuracy, precision, and recall. The ability of NLP to extract hidden information from textual data like app descriptions and reviews enables Ensemble machine-learning models to identify complicated malware patterns. However, a thorough knowledge of performance variances between various ensemble designs is required. Ensemble analysis is a dependable method for detecting Android malware, according to previous research. Yet, Q3 finds that malware may be successfully detected using ensemble analysis approaches, but certain studies, like [11] and [17], give performances poorly. It implies that ensemble analysis for malware detection remains neither reliable nor comprehensive.

Security: The discussion centers on security issues, highlighting the vital part that ensemble machine learning model integration plays in the field of Android malware detection. By enhancing the capacity to recognize harmful patterns, NLP provides more resilient protection against ever-evolving threats. Study [28] adds their extrinsic random-based ensemble method to the security conversation. This method demonstrates the ability to identify Android malware more accurately and robustly by combining many models. Furthermore, the study [29] explores the security environment with an emphasis on harmful domain identification. Their review, which focuses on NLP techniques, emphasizes how important language analysis is for spotting possible risks coming from malware domains.

Trade-offs: Whereas the methodologies under discussion present promising methods for detecting Android malware, we are subject to important trade-offs between practicality, comprehensiveness, and accuracy. When we talk about Accuracy vs. Code Obfuscation, non-ensemble methods struggle with Code Obfuscation, impacting accuracy. Whereas ensemble methods [8,10,16] are capable of extracting features from descriptions, the quality and comprehensiveness of the textual data may have an impact on how effective these methods are. App availability vs. Source code reliance is another trade-off, only a smaller selection of applications typically free ones has tools that need to access an app's source code [9]. Due to missing source code, there is a trade-off between having greater applicability across both free and premium apps and losing some analytical depth [20]. Overall vs. Breakdown metrics is another trade-off where authors reveal the overall accuracy of their models, hiding the performance differences across various malware categories [11]. There is a trade-off between openness and simplicity here because disclosing accuracy and precision breakdowns for each category can help better direct future developments [16]. Free vs. Paid Apps are another important trade-off for this research. Limited datasets to free applications ignore the variety of paid apps with various coding styles that are available in the Android ecosystem [36]. This means that there must be a trade-off made between easily accessible data and maybe overlooked targeted malware that may be contained in premium apps.

Weakness: The weakness of this paper's validity includes a collection of studies, data extraction and data analysis and the summary of the results.

Collection of studies using NLP: a compilation of research. Although we have made every effort to include pertinent research from journals and conferences across 5 databases in the above paper, certain pertinent papers may still be absent. Another issue associated with the collection of studies is the possibility of a few mistakes while sorting the research based on inclusion or exclusion criteria. We examine the list of publications using the cross-checking approach in the primary research to further prevent these mistakes.

Data extraction and data analysis: Due to the significant burden associated with data extraction and analysing, we additionally gather our data using the cross-checking approach. The terminal data is received whenever we reach a consensus over the comparison's outcomes. However, throughout the extraction and analysis of data, we still have a chance to commit some mistakes. Moreover, the original authors of the primary research should confirm this data to minimize errors.

Summary of the results: The factual accuracy of the findings in Q3 and Q4 is under question. There may not be definitive conclusions because of the variations in the research that were utilized as a comparison. Therefore, to minimize the variations in the prior analysis, we suggest creating a universal platform. Furthermore, to get specific and broad findings, further research on Android malware detection using ensemble analysis has to be gathered.

Challenges: Many challenges occur when developing malware identification tools. Firstly, the malware code obfuscated using anyone at all type of methods, causing issues with program comprehension and static type of analysis. Moreover, other applications consist of encrypted code, that is not available in the identification procedures. This happened with the DroidKungFu virus. This malware utilizes a separate key for each object of a program. A virus called Metamorphic may reform the names of the app's methods and classes by independently rewriting its code with each infection. The BFEDroid [8] was ineffective in analyzing this kind of malware. Moreover, numerous technologies [10,14,26] rely on access to an app's source code for analysis. Due to the lack of often available source code for apps, this typically prevents the tools from being applied. In truth,

only free apps have tested the latter; Another tool, like SEDMDroid [33], develops models to access just a finite sample of malware; in fact, only free applications have tested the latter. The lack of malware in a detection approach's training atmosphere can affect its efficacy. The primary advice is to remember that both harmful and benign applications may be used to test malware upon genuine datasets. Furthermore, there were different types of malware (adware, clones, data miners, etc.) that should be included in the datasets. Techniques may be more successful over a single form of malware while being less successful over multiple types of malware. Instead of using an overall number for the whole set of testing, researchers might benefit by addressing a breakdown of the fidelity and precision of their malware identification and detection approach. Lastly, both free and paid applications should include datasets, since the Android ecosystem stores the two kinds, and concerning the features of coding, they store both complimentary and commercial applications are probably going to differ greatly.

Testing datasets: Since various projects utilize various testing datasets, it is difficult to evaluate the efficiency, accuracy, and rates of false positives of distinct methodologies on an equal basis. According to primary studies, datasets are categorized into two groups based on their origin: in-lab datasets and in-the-wild datasets. In-lab datasets, which mostly comprise 3 datasets: Drebin, Gfan and Genome and in the wild include 4 datasets: Virusshare, Google Play, Appchina and Androzoo. We use the 3 most common datasets Drebin, Androzoo and Virusshare to compare the overall performance of ensemble models. This paper suggests that scholars choose between developing their own datasets or using existing publicly accessible datasets of Android applications, such as Androzoo or Derbin. In every other module, simply mention the year when the data set was published Also specify whether the database contains malware applications, harmless apps, or either malware and harmless apps.

4 Conclusion

This study offers a thorough review of Android malware detection by ensemble analysis, summarizing the most recent methods. In specific terms, 30 investigations from 2019 to 2023 carry out this SLR. Additionally, this SLR looks at the many types of ensemble analysis techniques, the method of performing empirical tests, the capacity of ensemble analysis approaches to identify malware and the effectiveness of various models in the detection of Android malware.

This review focused on a multiple number of techniques for Android malware detection, that target robustness and weakness for every research and advise about further investigation upon this issue. This review makes some recommendations that allow researchers to create more effectual and efficient malware detection tools. This review discovers that: (1) the foremost widely used ensemble analysis method for detecting Android malware is the static ensemble method; and (2) the majority of the datasets used in empirical tests are experimental datasets like Drebin, Androzoo and virus share. Scikit-learn is used by the majority of research as an ensemble analysis support tool. The most popular features are static ensemble features and hybrid ensemble features Out of all the models that are utilized, the machine learning model has the most share. The most often used performance metric is accuracy; (3) Empirical data indicates that ensemble analysis methods are useful for detecting malware; (4) Through the examination and comparison of primary research, it infers early findings that the ensemble model performs better than the non-ensemble model. Future prospects for more studies related to this field are presented as follows:

- The current technique has to be reevaluated and re-implemented with a liberated, cutting-edge dataset. It's critical to comprehend where those techniques stack up against one another, and also initiate whether these techniques are applicable for detecting existing malicious threats.
- With the constant growth of Android, the latest and most effective tools are required. The present release of Android does not support the existing tools.
- The development of updated datasets to utilize with the advancement of detection of Android malware programs also requires the usage of automated tools.
- Most research in this field makes use of obsolete datasets that are no longer suitable for the present framework of the Android operating system. This review specifically suggests that the testing datasets used by the researchers are accessible to others, along with the analysis they perform. Since multiple testing sets of data are typically employed and different identification methods are disclosed, it has proven particularly challenging to differentiate between them.
- Better results announcement skills are required, along with the use of a wider variety of measures.

Thus, the development in this area is inspiring, according to the literature, ensemble machine-learning algorithms performed better than traditional identification of malware approaches. The continuous evolution of newly created machine-learning techniques indicates that there is still plenty of room for future study in that sector.

5 Acknowledgement

This research was conducted without external funding support. The software used for data analysis and computation was Python.

References

1) Chowdhury MN, Haque A, Soliman H, Hossen MS, Fatima T, Ahmed I. Android Malware Detection using Machine learning: A Review. 2023. Available from: https://doi.org/10.48550/arXiv.2307.02412.
2) Mijoya IB, Khurana S, Gupta N. Malware detection in Android devices Using Machine Learning. In: 2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), 04-05 November 2022, Greater Noida, India. IEEE. 2023;p. 307–312. Available from: https://doi.org/10.1109/ICCCIS56430.2022.10037699.
3) Rani S, Tripathi K, Kumar A. Machine learning aided malware detection for secure and smart manufacturing: a comprehensive analysis of the state of the art. *International Journal on Interactive Design and Manufacturing (IJIDeM)*. 2023;p. 1–28. Available from: https://doi.org/10.1007/s12008-023-01578-0.
4) Muzaffar A, Hassen HR, Lones MA, Zantout H. An in-depth review of machine learning based Android malware detection. *Computers & Security*. 2022;121:1–21. Available from: https://doi.org/10.1016/j.cose.2022.102833.
5) Agrawal R, Shah V, Chavan S, Gourshete G, Shaikh N. Android Malware Detection Using Machine Learning. In: 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), 24-25 February 2020, Vellore, India. IEEE. 2020. Available from: https://doi.org/10.1109/ic-ETITE47903.2020.491.
6) Pradeepa G, Devi R. Malicious Domain Detection using NLP Methods — A Review. In: 2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART), 16-17 December 2022, Moradabad, India. IEEE. 2023;p. 1584–1588. Available from: https://doi.org/10.1109/SMART55829.2022.10046882.
7) Sen S, Can B. Android security using nlp techniques: a review. 2021. Available from: https://doi.org/10.48550/arXiv.2107.03072.
8) Chimeleze C, Jamil N, Ismail R, Lam KY, Teh JS, Samual J, et al. BFEDroid: A Feature Selection Technique to Detect Malware in Android Apps Using Machine Learning. *Security and Communication Networks*. 2022;2022:1–24. Available from: https://doi.org/10.1155/2022/5339926.
9) Ding Y, Zhang X, Hu J, Xu W. Android malware detection method based on bytecode image. *Journal of Ambient Intelligence and Humanized Computing*. 2023;14(5):6401–6410. Available from: https://doi.org/10.1007/s12652-020-02196-4.
10) Shatnawi AS, Yassen Q, Yateem A. An Android Malware Detection Approach Based on Static Feature Analysis Using Machine Learning Algorithms. *Procedia Computer Science*. 2022;201:653–658. Available from: https://doi.org/10.1016/j.procs.2022.03.086.
11) Amer E. Permission-Based Approach for Android Malware Analysis Through Ensemble-Based Voting Model. In: 2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), 26-27 May 2021, Cairo, Egypt. IEEE. 2021;p. 135–139. Available from: https://doi.org/10.1109/MIUCC52538.2021.9447675.
12) Alzahrani AIA, Ayadi M, Asiri MM, Al-Rasheed A, Ksibi A. Detecting the Presence of Malware and Identifying the Type of Cyber Attack Using Deep Learning and VGG-16 Techniques. *Electronics*. 2022;11(22):1–20. Available from: https://doi.org/10.3390/electronics11223665.
13) Congyi D, Guangshun S. Method for Detecting Android Malware Based on Ensemble Learning. In: ICMLT '20: Proceedings of the 2020 5th International Conference on Machine Learning Technologies. ACM. 2020;p. 28–31. Available from: https://doi.org/10.1145/3409073.3409084.
14) Bhat P, Behal S, Dutta K. A system call-based android malware detection approach with homogeneous & heterogeneous ensemble machine learning. *Computers & Security*. 2023;130:103277. Available from: https://doi.org/10.1016/j.cose.2023.103277.
15) Jing P, An N, Yue S. Dynamic detection method for Android terminal malware based on Native layer. In: 2023 4th International Conference on Computer Engineering and Application (ICCEA), 07-09 April 2023, Hangzhou, China. IEEE. 2023;p. 83–86. Available from: https://doi.org/10.1109/ICCEA58433.2023.10135186.
16) Islam R, Sayed MI, Saha S, Hossain MJ, Masud MA. Android malware classification using optimum feature selection and ensemble machine learning. *Internet of Things and Cyber-Physical Systems*. 2023;3:100–111. Available from: https://doi.org/10.1016/j.iotcps.2023.03.001.
17) Zhou H, Zhang S, Yong F, Pan H, Guo W. An Android Malware Detection Approach Based on Summation of Multi-order Derivatives LSTM. Research Square Platform LLC. 2022. Available from: https://doi.org/10.21203/rs.3.rs-2337299/v1.
18) Karbab EB, Debbabi M. MalDy: Portable, data-driven malware detection using natural language processing and machine learning techniques on behavioral analysis reports. *Digital Investigation*. 2019;28(Supplement):S77–S87. Available from: https://doi.org/10.1016/j.diin.2019.01.017.
19) Sharan AS, Radhika KR. Machine Learning Based Solution for Detecting Malware Android Applications. *International Journal of Innovative Research in Applied Sciences and Engineering*. 2020;4(3):664–668. Available from: https://www.ijirase.com/assets/paper/issue_1/volume_4/V4-Issue-3-664-668.pdf.
20) Xu P. Android-COCO: Android Malware Detection with Graph Neural Network for Byte- and Native-Code. 2021. Available from: https://doi.org/10.48550/arXiv.2112.10038.
21) Abualghanam O, Alazzam H, Qatawneh M, Aladwan O, Alsharaiah MA, Almaiah MA. Android Malware Detection System Based on Ensemble Learning. Research Square Platform LLC. 2023. Available from: https://doi.org/10.21203/rs.3.rs-2521341/v1.
22) Kuchipudi R, Uddin M, Murthy TS, Mirrudoddi TK, Ahmed M, Babu PR. Android Malware Detection using Ensemble Learning. In: 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS), 14-16 June 2023, Coimbatore, India. IEEE. 2023;p. 297–302. Available from: https://doi.org/10.1109/ICSCSS57650.2023.10169578.
23) Zhu HJ, Li Y, Wang LM, Sheng VS. A multi-model ensemble learning framework for imbalanced android malware detection. *Expert Systems with Applications*. 2023;234:120952. Available from: https://doi.org/10.1016/j.eswa.2023.120952.
24) Zhang N, Xue J, Ma Y, Zhang R, Liang T, an A Tan Y. Hybrid sequence-based Android malware detection using natural language processing. *International Journal of Intelligent Systems*. 2021;36(10):5770–5784. Available from: https://doi.org/10.1002/int.22529.
25) Aurangzeb S, Aleem M. Evaluation and classification of obfuscated Android malware through deep learning using ensemble voting mechanism. *Scientific Reports*. 2023;13(1):1–12. Available from: https://doi.org/10.1038/s41598-023-30028-w.
26) Atacak İ. An Ensemble Approach Based on Fuzzy Logic Using Machine Learning Classifiers for Android Malware Detection. *Applied Sciences*. 2023;13(3):1–26. Available from: https://doi.org/10.3390/app13031484.
27) Ullah F, Ullah S, Srivastava G, Lin JCW. Droid-MCFG: Android malware detection system using manifest and control flow traces with multi-head temporal convolutional network. *Physical Communication*. 2023;57:101975. Available from: https://doi.org/10.1016/j.phycom.2022.101975.

28) Mahindru A, Sangal AL. MLDroid—framework for Android malware detection using machine learning techniques. *Neural Computing and Applications*. 2021;33(10):5183–5240. Available from: https://doi.org/10.1007/s00521-020-05309-4.
29) Potha N, Kouliaridis V, Kambourakis G. An extrinsic random-based ensemble approach for android malware detection. *Connection Science*. 2021;33(4):1077–1093. Available from: https://doi.org/10.1080/09540091.2020.1853056.
30) Rana MS, Sung AH. Evaluation of Advanced Ensemble Learning Techniques for Android Malware Detection. *Vietnam Journal of Computer Science*. 2020;07(02):145–159. Available from: https://doi.org/10.1142/S2196888820500086.
31) Yang Y, Du X, Yang Z, Liu X. Android Malware Detection Based on Structural Features of the Function Call Graph. *Electronics*. 2021;10(2):1–17. Available from: https://doi.org/10.3390/electronics10020186.
32) Kim M, Kim D, Hwang C, Cho S, Han S, Park M. Machine-Learning-Based Android Malware Family Classification Using Built-In and Custom Permissions. *Applied Sciences*. 2021;11(21):1–24. Available from: https://doi.org/10.3390/app112110244.
33) Zhu H, Li Y, Li R, Li J, You Z, Song H. SEDMDroid: An Enhanced Stacking Ensemble Framework for Android Malware Detection. *IEEE Transactions on Network Science and Engineering*. 2021;8(2):984–994. Available from: https://doi.org/10.1109/TNSE.2020.2996379.
34) Abubaker H, Ali A, Shamsuddin SM, Hassan S. Exploring permissions in Android applications using ensemble-based extra tree feature selection. *Indonesian Journal of Electrical Engineering and Computer Science*. 2020;19(1):543–552. Available from: http://doi.org/10.11591/ijeecs.v19.i1.pp543-552.
35) Guan S, Li W. EnsembleDroid: A Malware Detection Approach for Android System based on Ensemble Learning. In: 2022 IEEE MIT Undergraduate Research Technology Conference (URTC), 30 September 2022 - 02 October 2022, Cambridge, MA, USA. IEEE. 2023;p. 1–5. Available from: https://doi.org/10.1109/URTC56832.2022.10002213.
36) Amer E. Permission-Based Approach for Android Malware Analysis Through Ensemble-Based Voting Model. In: 2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), 26-27 May 2021, Cairo, Egypt. IEEE. 2021;p. 135–139. Available from: https://doi.org/10.1109/MIUCC52538.2021.9447675.
37) Mehtab A, Shahid WB, Yaqoob T, Amjad MF, Abbas H, Afzal H, et al. AdDroid: Rule-Based Machine Learning Framework for Android Malware Analysis. *Mobile Networks and Applications*. 2020;25(1):180–192. Available from: https://doi.org/10.1007/s11036-019-01248-0.