Asian Journal of Research in Computer Science

11(3): 44-60, 2021; Article no.AJRCOS.73749
ISSN: 2581-8260

A State of Art Survey for Understanding Malware Detection Approaches in Android Operating System

Suhaib Jasim Hamdi[1*], Naaman Omar[1], Adel AL-zebari[1], Karwan Jameel Merceedi[1], Abdulraheem Jamil Ahmed[1], Nareen O. M. Salim[1], Sheren Sadiq Hasan[1], Shakir Fattah Kak[1], Ibrahim Mahmood Ibrahim[1], Hajar Maseeh Yasin[1] and Azar Abid Salih[1]

[1]Duhok Polytechnic University, Duhok, Kurdistan Region, Iraq.

Authors' contributions

This work was carried out in collaboration among all authors. All authors read and approved the final manuscript.

Article Information

DOI: 10.9734/AJRCOS/2021/v11i330266
Editor(s):
(1) Dr. Hasibun Naher, BRAC University, Bangladesh.
Reviewers:
(1) Robiah binti Yusof, Universiti Teknikal Malaysia Melaka, Malaysia.
(2) Pasupuleti Venkata Siva Kumar, Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering &Technology, India.
Complete Peer review History: https://www.sdiarticle4.com/review-history/73749

Review Article

Received 01 July 2021
Accepted 06 September 2021
Published 07 September 2021

ABSTRACT

Mobile malware is malicious software that targets mobile phones or wireless-enabled Personal digital assistants (PDA), by causing the collapse of the system and loss or leakage of confidential information. As wireless phones and PDA networks have become more and more common and have grown in complexity, it has become increasingly difficult to ensure their safety and security against electronic attacks in the form of viruses or other malware. Android is now the world's most popular OS. More and more malware assaults are taking place in Android applications. Many security detection techniques based on Android Apps are now available. Android applications are developing rapidly across the mobile ecosystem, but Android malware is also emerging in an endless stream. Many researchers have studied the problem of Android malware detection and have put forward theories and methods from different perspectives. Existing research suggests that machine learning is an effective and promising way to detect Android malware. Notwithstanding, there exist reviews that have surveyed different issues related to Android malware detection based on machine learning. The open environmental feature of the Android environment has given Android an extensive appeal in recent years. The growing number of mobile devices, they are

**Corresponding author: E-mail: Suhaibbaroshky@gmail.com;*

incorporated in many aspects of our everyday lives. In today's digital world most of the anti-malware tools are signature based which is ineffective to detect advanced unknown malware viz. Android OS, which is the most prevalent operating system (OS), has enjoyed immense popularity for smart phones over the past few years. Seizing this opportunity, cybercrime will occur in the form of piracy and malware. Traditional detection does not suffice to combat newly created advanced malware. So, there is a need for smart malware detection systems to reduce malicious activities risk. The present paper includes a thorough comparison that summarizes and analyses the various detection techniques.

Keywords: Malware; detection; operating system; android; viruses.

1. INTRODUCTION

Android is the most popular smartphone platform in today's market, and its popularity is growing by the day, Malware includes viruses from computers, worms, backdoors, spyware, Trojans and other harmful systems [1]. There are many malware strategies that target the Android platform without the victim's awareness by transferring confidential information [2].The Android operating system is usually regarded as the most popular and regularly affected [3]. The quick growth of the mobile Internet has made Android the smartest terminal operating system in the world, Mobile malware has become a serious cyber security problem [4]. The phrases 'virus' and 'malware' are often used yet vary technically. Malware is a comprehensive phrase including all kind of malware, by accessing the infected folder or application, the victim is woken up to the infection [5]. The virus might erase or encrypt the data while it is running the infection [6]. In addition, the software may be changed or system features may be disabled, the software detection system focuses on the qualities both of the execution and of the source code of the program, the software detection system focuses on the qualities both of the execution and of the source code of the program Android malware detection techniques to machine learning [7]. Malware analyses are a procedure that detects software programs to determine their behaviours, functioning and whether or not they are malware. Methods of Android malware detection may be classified as static and dynamic analysis [8]. That indicates that every 10 seconds a new Android malware application was identified, Malware detection technique may be classified as static detection, dynamic detection and hybrid detection in three categories [9]. Since android is the most common operating system used for Internet access. Android includes an operating system, an app framework and key apps [10]. Each Android app is segregated from other applications [11]. Machine Learning algorithms and methods have reached a high accuracy in malware detection among the several methods in the detection of malware [12]. Many mobiles with several operating systems are available. Android is a mobile open-source operating system that can be accessed on numerous devices. Android devices are activated every day according to Google 1.3 million, the risk of malware will rise by extending mobile phone capabilities [13].

2. ANDROID MALWARE

Android OS has become a major malware target, because it is popular. In August 2010, Kaspersky Lab Researcher Dennis Mashlennikov uncovered and revealed the first known Android virus utilizing text messaging [14]. The virus masks as a film player and delivers SMS messages to two top quality phone lines without the owner's awareness. The cost of each communication is $5, which leads in premium message service owners owing considerable money [15]. Russian users alone and the virus on Android Market was not found (which was renamed to Google Play in 2012).

During the same month in January, Symantec disclosed a GPS tracking program that captured and relayed position code to a remoten server every 15 minutes without user awareness' [16]. While it was reported that it was discovering the first Android SMS virus in this month. Due to the failure of the virus on the Android Market at that time, the harm was limited. Lookout stated in the final quarter of 2010 that the latest Android virus was found in the wild. An example by Geinimi demonstrated the "Trojanization" idea in the Android environment [17].

More "Trojanized" versions of authentic applications have been identified and reported according to statistics provided in 2011 [17]. The golden rule for Android users was that only

applications from the official Android market should be downloaded and installed [1]. But since Lookout discovered more than 50 fraudulent Android applications on the Android official market, this advice has become less useful [18].

In 2010 and 2011 the malware outbreak of Android began with the increase in the number of security malware detections [19]. Only 0.5 percent of the overall mobile malware made up by Android malware in 2010, Juniper Networks said [20]. Over the course of a year, Android's malware threats had risen to 46.7%, while Android malware had reached 47% of all mobile malware risks by the end of 2011 [21]. By March 2013, all mobile malware risks were reached by Android. Nearly every virus targeted at Android systems in 2013, as demonstrated in February 2014 by the Kaspersky Lab [22].

As Android malware is increasing, academic and industry researchers have focused on studying this expanding danger [23]. Several researchers concentrated on collecting Android samples, meticulous analysis of the data set and description of the results gathered in order to get a knowledge of Android malware attack pathways and infection behaviours [24]. Researchers have also set up numerous tools and methods to support the analysis of Android malware. Droid-Scope is an example of a platform built in a virtualized environment to analyse malware [25].

There have also been a lot of studies about Android malware identification and prevention; several technologies have been created to help screen Android applications and identify possible harmful ones [26]. This is a case in point with Risk Ranker, a software tool meant to assess if a program is hazardous (e.g., launching a root exploit or sending background SMS messages) [27].

3. ANDROID MALWARE DETECTION

The analysis according maybe classified to the functionality used to characterize an application as static and dynamic. Static analysis without program execution is carried out [28]. To categorise them, the API calls can be located in the AndroidManifest.xml file and I separated them into static features like permissions and dynamic functions [29]. The characteristics of a program found while in operation include features to be analyzed dynamically, such as network traffic, battery use, IP address, and more [30]. A mix of static and dynamic analytical techniques is hybrid analysis. The following sections discuss the characteristics and algorithms used to develop this specific app [31].

3.1 Static Analysis

Static analyses are used to verify the structure of the program without running it. This is the safest way to test malware, because if you execute code, it will infect your system [32]. While its most basic version looks for malware information without looking at the code, static analysis reveals this information [29]. The name of malware files, types of files, and file sizes might give suggestions as to the virus [33]. A comparison between MD5 scans or Hashes and a database may be made to see whether the virus has been found before. A scan of antivirus software will show you which malware you work on, as well as screening your machine for infections [32].

Advanced static analysis called code analysis, which differs from codes, dissects the binary file to scrutinize all its distinct components [34]. A disassembler is used to reverse engineer the source code [35]. The code of the machine is turned into a legible code for human assembly The function of a program may be inferred by viewing the mounting instructions [36]. A file can include essential information about the headers, functions and strings. But modern hackers can avoid this countermeasure quickly [37]. This approach requires grammatical errors to be included in your code so that disassemblers do not recognize the malicious code [38]. Dynamic malware analysis is necessary to circumvent the static malware analysis [39].

Methods of static analysis seek for specified characteristics saved in the program. This method is time- and resource efficient because the program is not carried out [40]. The study nevertheless faces the coding strategies of the developers of malware which allow them to escape static detection [41]. The most important thing to accomplish is to download an update, after which the malicious software is loaded, by installing a genuine program on a mobile device. Only the benign application is scanned by static analysis; thus, this is not discoverable [42].

The most commonly used are these static features: Permits and calls to the API [43]. These functions have been studied and investigated in

depth and in order to assist establish whether a specific malware piece is harmful, together with the extracted metadata accessible in the Google Play Store, such as the version name, version number, the author's name, the most updated time, etc [44].

3.2 Dynamic Analysis

Dynamic analytics are sometimes referred to as malware conduct analysis. In a secure environment malicious software must be analysed [45]. A sandbox environment is built within a virtual network segregated from the rest of the network to build malware without compromising its actual systems [42]. The analysis is feasible without any long-term repercussions and afterwards the sandbox might be reinstated to its original condition [46].

When a malware piece is run and its detection signature can be identified by dynamic analysis, the technical indications are issued [47]. A sandbox-controlled program is called a dynamic analytics software to identify the effects of malware. The updates may include new registry keys, IP addresses, domain names, and file path locations [48]. It will also indicate whether the infection is connecting to an external server controlled by a hacker through the application of dynamic analysis [25]. It is both a helpful and time-consuming dynamic analytical approach. During execution of the malware, a debugger can focus on each behaviour of the software while processing the instructions [49,50].

Like static analysis, fraudsters are struggling to detect and film dynamic analyses. If malware suspects a virtual setting or a debugger, it won't work [51]. There might be a delay between launching and executing the risky payload of the application or specific user input [52]. The most thorough knowledge of a malware threat is often the application of static and dynamic analytical approaches in combination [53].

4. MALWARE DETECTION TECHNIQUES

Malware methods are used to identify malicious software and prevent computer system infection, therefore preventing it from losing potential information [54]. Abandonment of the system. Three ways to detect and categorize malware are available:

- Detection on permission basis.
- Detection on the basis of signature
- Methods based on pacification [11].

4.1 Malware Based on Machine Learning Android Malware Schemes

Application and application of machine learning, which has been defined by different academics, is an artificial intelligence research branch [55]. Machine learning, according to, comprises of a series of approaches for automating predictions on the basis of historical data [56]. The machine learning may be separated into five paradigms with various theoretical notions based on a comparison between the master's learning algorithms and activities carried out by the human brain: symbolists, connectivity, evolutionist, Bayesian and analogizer. Each machine learning category has its own fields of study and algorithms [57,58].

4.2 Android Malware's Risks

Once a malware-infected Android OS is installed, users are exposed to and are exposed to several risks [59]. A some of the Include but not limited to: the dangers they may suffer:

- Database loss
- Theft of personal data that may lead to Dieft of identity
- Users' spying
- Telephone remote operation
- Ransomware financial loss [60].

To detect the unknown malware using machine learning technique, a flow chart of their approach is shown in Fig. 1. It includes pre-processing of dataset, promising feature selection, training of classifier and detection of advanced malware [61].

4.3 Random Forest Algorithm

The random forest provides random selection attributes based on decision-making trees [62]. The classic decision tree selects an excellent attribute in the current node attribute set [63]. The random forest picks a subset of the k characteristics in the set of node characteristics randomly. And then chooses an ideal attribute for the partition from the subset [64].

4.4 Android Malware Based: Machine Classification Algorithm Android Malware

Classification Machine learning the topic of machine learning involves theory and statistics of

probability. The learning model of the machine is primarily designed to provide algorithms that allow the computer to learn [65].

Static Analysis: Static analysis is used to decompile the application to retrieve the code file and extract the features without executing the application software via reverse engineering. These characteristics are static [66].

Dynamic Analysis: The dynamic analysis approach is to replicate the behaviour of the user by executing the program and to identify if malicious software is based on the program's real functioning [67].

4.5 Permission Based Analysis

In determining which apps have access to resources, application permissions are essential [68]. Most programs have no user data access and have no default affect system safety [69]. The user must permit the app to access all the required resources at installation time [70]. It is a good idea for developers to clarify the rights requested for resources in order to give transparency [71]. However, as shown, not all permissions are necessary [72]. The permission-based detection is good for the speed of malware screening but should not be utilized on other files containing the hazardous code [73]. In addition, allowance between malicious and benign apps is slightly different, thus a second pass requires permission-based detection [74,75].

4.6 Virtual Machine Analysis

A virtual machine is used to test byte code for a specific application [76]. Tests application behaviour, and tests for hazardous aspects, such as data flow and control flow, that may aid in malevolent apps [77]. Most virtual apps for mobile devices have already been deployed, in particular for Android. By tracing API calls, DroidAPIMiner identifies malware that might lead to the disclosure of sensitive data [78]. The fact that it is done at the instruction level, which takes up more processing power and memories, is also an issue with virtual analytics [79].

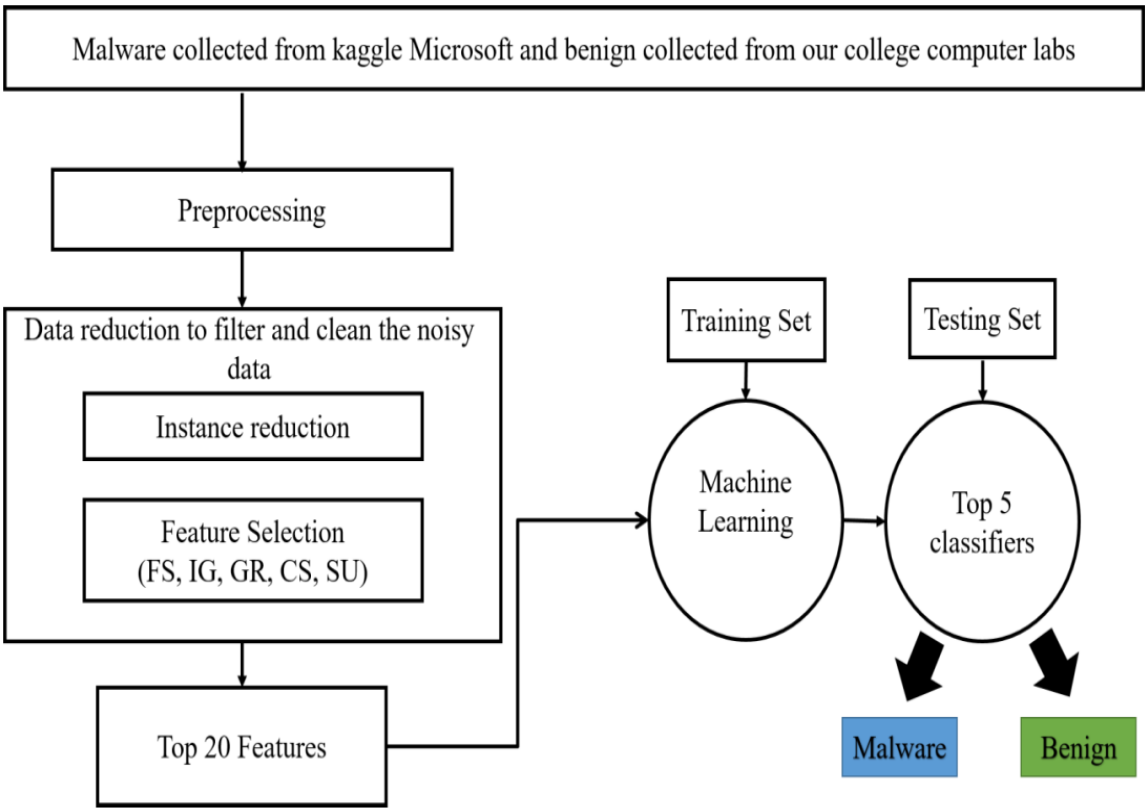

Fig. 1. Malware detection flowchart [61]

4.7 Anomaly Based

Analysis based on anomalies is based on monitoring device behaviour, tracking the various parameters and state of the device's components [80]. A malware detection approach based on behaviour. Randomly checks several characteristics of the device status including battery level, CPU use, network traffic, etc to identify a malware. Measuring is done during operation and then delivered to an algorithm which classifies it properly [81,82]. The two distinct anomaly-based technologies used to identify malware in Android devices are CrowDroid and AntiMalDroid [83]. The first is based on the analysis of system calls logs, which analyses an application's activity then creates malware signatures [84]. SMS Profiler and iDMA are two instruments used to unlawfully identify the use of iOS services [85].

4.8 Taint Analysis

Taint droid is a software program that collects sensitive information from many places and identifies private data leakage in mobile applications [86]. The tool tags and tracks sensitive data while moving from the device in order to protect against misuse of data. Control flow tracking is not implemented while efficient data tracking is given [87].

5. ALGORITHMS FOR MACHINE LEARNING

There is an enormous range of classifiers that can be utilized for machine learning [88]. Once the current techniques to machine learning have been intensively studied, the downsides and advantages of succeeding algorithms have been highlighted such that I consider that I am particularly willing to be able to identify malware:

K-Nearest Neighbour (knn): Although it is claimed to be an extremely simple algorithmic program (silent algorithms) and performs quickly, it is improper or not very rewarding as soon as the training set is blare or outliers undergoing.

Subject Vector Machine Support (SVM): method comprises a robust and intricate theoretical and abstract basis, since it typically performs more than alternative algorithms for classification outcomes.

Decision Tree (J48): might be a classification tree which hopes to categorize the instances properly with functional values. There are nodes and distribution leaves in a decision tree.

Neural networks (NN): is another extremely productive, human-brain-based machine learning method. Neural networks methodology is nevertheless longer than alternative clinicians, and is considered to be troubling or rigorous, in which real time might be limiting in any malware detection system.

Naive Bayes (NB): presume that the structures are casual sovereigns and calculate its potential for the decision to be appealed [89].

6. LITERATURE REVIEW

Over the last few years, Android is considered as the most commonly adopted OS and has drawn attention to the malware maker as a result of its increasingly growing success. Android enables apps to be downloaded and installed from other unofficial markets. Existing study obviously lags behind in the effective and precise detection of malware. An effective and accurate solution to this problem provided, called SAMADroid, a new 3-degree hybrid malware detection model for Android operating systems, SAMADroid is a new malware identification model that combines the advantages of static analysis, dynamic analysis, and intelligence learning. Based on the advantages and disadvantages of current anti-malware technologies [3]. Many attacks targeting Android phones may be carried out, mostly by the development of applications. Some classification algorithms have been evaluated in their research to assess best performance. Algorithm when it comes to malware identification android. An Android device data collection was collected from fig share and used for information in the Waikato environment, Training and research analysis (WEKA), calculated by accuracy, false-positive rate, accuracy, retrieval, f-method, receiver operating curve (ROC) and root-mean square, Mistake (RMSE). Multi-layer perceptron’s were found to work best with 99.4 percent accuracy, their project was designed to test Android malware classification algorithms [60]. A security detection approach based on the Metropolis algorithm is proposed in their article on Android introduce a concept method named PPMDroid to conserve bandwidth and speed up the process with many optimizations [90]. Today, in all countries, the usage of cell phones is increasing and sadly, cyber criminals are constantly targeting mobile phones. The key cause of this kind of assault is

the malicious software that a consumer downloads from reputable media like Play store, the App Store and everything. Their framework is a smartphone android technology focused on deep learning. In order to detect the malicious actions of an algorithm the application can conduct static and dynamic analysis. The method will combine both static and evolving effects of study. When security makers used signature detection to detect a ransomware, attackers began to create a new signature to circumvent those solutions. This reduced the reliability of those solutions apps [91]. This approach evaluates the 24 risky allowances of Android using the Metropolis algorithm; Removes permissions for uncertainty, extracts characteristics of permission. Their solution decreases the detecting function and can achieve 93.5 percent for the accuracy of harmful program detection [64]. Their paper is built on Android learning systems and machinery. It identifies Android malware from two static and dynamic analytical perspectives.

Using machine learning, it is feasible to successfully identify malware with Android malware. The combination of static analysis and dynamic analysis may simultaneously increase detection accuracy and efficiency. Fig. 2 represents the Architecture of Droid Deep [67]. Their research condenses the development of malware-detection technology that supports machine study algorithms based on Android operating systems. In their report, they suggested a portable malware position display to speed up the efficiency of operation classifier with 9 movement highlights. The model also uses grouping techniques like stream, package and time-based highlights to describe families of malware. Mobile malware is thus pernicious and therefore it is essential for users to provide a fast and accurate detection method, Minimize malware investigation costs by picking representative samples only 8.5 % to 22 percent [89]. In their article they suggested a malware identification scheme to safeguard the protection of Android that protects the privacy (or assets) of telephone vendors, consumers and security service providers. It identifies malicious applications in app stores of telephone providers and on phones of consumers without exchanging data. The privacy problems of current static and dynamic malware detection methods are highlighted; runtime actions of apps and malware signatures with others. Tey suggested DL-Droid, a deep learning framework used in state-of-the-

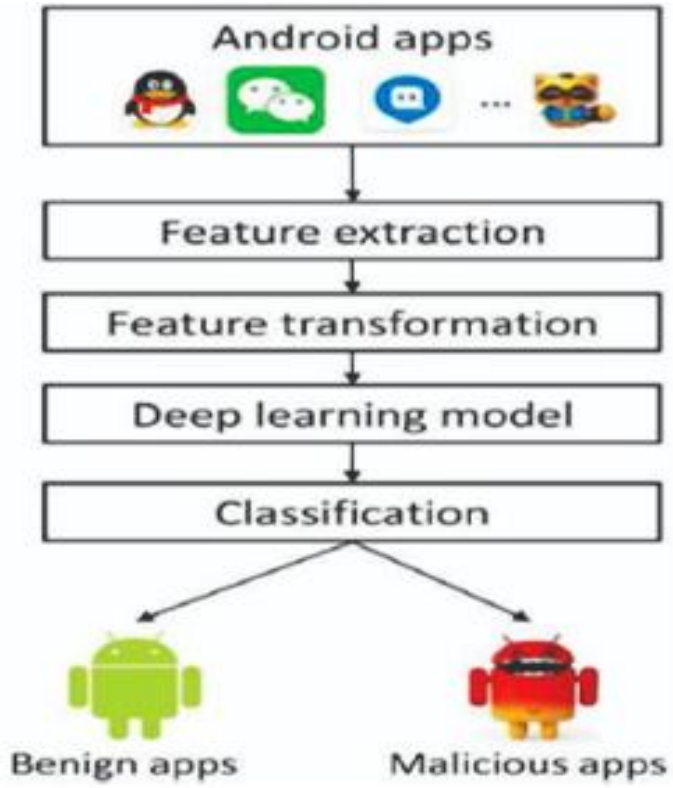

Fig. 2. Architecture of droid deep [57]

art input generation to detect malicious apps from Android. Experiments on actual devices per developed with more than 30,000 applications (benign and malware). In their article, they introduced DL-Droid, an advanced dynamic analysis system for the identification of Android malware, Droid uses profound learning with a standard input generation technique, although it has the potential to use the popular Monkey platform in state-of-the-art practice (stateless method). This is the first study to examine profound learning with complex functions derived from smartphones utilizing actual mobile. Their findings also emphasized the importance of enhanced input generation for complex analytical systems built to identify Android malware through machine learning .Today, several smartphone operating systems are used, including various formats and market shares. Mobile networks, like other information systems, are sensitive to virus assault. Detection of malware is very critical and is a must-deliver method for protecting and minimizing private data in any system. They examined and reviewed various strategies of malware identification for mobile operating systems in their article. The objective of the paper is to create a user profiling method for the mobile identification of malware. The vulnerability of any malware identification strategy has already been noted and debated. They built a new malware identification platform, focused on a mobile user profiling, for mobile applications [92]. With the spread of Android-based smart Internet of Things (IoT), malicious Android apps for IoT devices have attracted the publicity because of their privacy and property loss concerns. Their paper introduced Eve Droid, a malware identification framework designed to allow detection of Android and IoT malware, which allows the IoT world safer by decreasing malicious software running on Android-equipped smartphones. Due to API changes, and now able to capture previously unseen activities. The results also demon state that EveDroid is more accurate and robust to malware evolvement compared to existing detection systems [93]. Classification utilizing machine learning was an important class of malware security solutions. Also proposed a new classification method based on the findings of a longitudinal analysis on Android applications focusing on their complex behaviour. The key lesson learned is that learning software development offers a promising way to identify malware over the long term, they have examined a new approach cantered on an evolutionary characterization of smartphone behaviours. These results showed the potential of long-term malware identification approaches focused on evolution [94]. Android is one of the main targets for attackers to unleash destructive intentions. Researchers each year propose a new Android malware analyser system to protect against Android malware apps in the field of real life. In the analysis they compared accessible Android malware datasets with 15 key requirements and identifying key weaknesses. In this portion, they proposed the second part of CICAndMal2017 which incorporates new feature sets such as static permissions and attempts and the appended API calls. In the portion of dynamics [95]. In recent years, many mobile malware detection systems have been suggested to deal with this issue. Their paper is about the survey of current smartphone detection systems for malware. Their paper includes a systematic investigation of some accurate structures built on a method of static analysis. It explains and evaluates each scheme. In addition, both programs are similar to a tutorial on techniques. Malware detection is considered a key precondition for Android Operating System to defend mobile users from personal theft of privacy [96]. Two kinds of characteristics, permission requests and system calls are examined as a technique to identify malware, in their study. By using a machine learning technique, they can distinguish between benign and malicious applications. The model used permissions to get an accuracy of approximately 80% and system calls to reach a classic cation accuracy of about 60%. Their article analysed two major characteristics for Android malware detection, permission and system calls, and applied machinery education to both. The results suggested that permission data is better for malware detection than system call data [97]. Their paper is based on Android and learning machines. It identifies malware from two static analytical and dynamic analytical aspects. With machine learning for Android malware detection, it is feasible to identify malware successfully. Static analysis and dynamic analysis may simultaneously increase detection accuracy and efficiency, to discover malicious software, a security technique must be developed. , The exponential growth in the amount of Android malware presents great challenges for malware programs because the number of malware samples is overwhelming. They built Android, a new framework that automatically classifies Android malware samples with high precision and accelerates malware detection efficiently by proposing representative malware screening samples. Android is more efficient and reliable

than advanced methods. Malware inspection and malware raising to avoid analysis. It offers significant knowledge for the identification and inspection of malware and increases malware levels to avoid review [98]. They discussed various forms of Android malware detection technology applying various methods of deep learning in their article. Due of Android's open nature, here they are investigating a number of various malware methods of detection such as: MalDozer, Droid Detector, Droid Deep Learner and Deep Flow. It employs a static analysis approach as well as an API technique. MalDozer is used to detect malware in the Convolution Neural Network, Whether or if an android application is contaminated with malware without a facility, Their goal was to create a profound learning model which can recognize automatically [99]. In order to train the pre-processed sequences, they next utilize two deep learning methods: DexCNN and DexCRNN. Two meth- abilities have been examined on a data set of 80 0 benign APKs and 80 0 malignant APKs, your study presents two detection approaches for end-to-end malware without human engineering. First, they utilized the sample retrieval technique to pre-process the classes.dex APK file. DexCNN can achieve accuracy of 93.4 percent, while the DexCRNN can reach accuracy of 95.8 per cent. Other comparable malware detection tasks may readily be expanded to the approaches given [100]. This mode of detection increases to some level the detection accuracy. The random forest method has been upgraded to yield flourishing sets. Then the approach is used Rules for sensitive authorization, to analyse this detection mechanism and validate the efficacy of the system, a number of assessment approaches have been applied. Utilized to assess the method of detection and to validate the system's efficacy [101]. DAMBA, a novel prototype system based on C/S architecture, is presented in their article. DAMBA extracts the application's dynamic and static characteristics. For further studies, they provided the TANMAD-method, a two-step methodology for detecting malware from Android, which minimizes the spectrum of probable families of malware. They provided numerous optimization ideas for hybrid analysis to achieve improved efficiency and precision in their papers. The complicated computation work of the PC customer was finished to maintain the limited resources of the mobile customer [102].Cloak & Dagger, a specific kind of assaulting action, is detailed in length. Detection algorithm for harmful software packages, The Cloak and Dagger attack algorithm is presented for the detection of malicious software packages. It is suggested that you conduct a Cloak and Dagger assault [103]. The detection will be measured using three distinct classifications: K-nearest (KNN), Random Forest (RF) and Decision Tree (DT). In the identification and classification of computer malware, a visualization methodology was used, although not many trials concentrated on Android operating system. By utilizing the Random forest machine learning method on picture characteristics created from APK samples, the suggested study could reach 84.14 percent detecting accuracy [104]. They used the technology to build a framework named the ONAMD Online Android Malware Detection Approach, The ONAMD initially collects the details (e.g., requested permissions, and basic data info, etc.). Next, the SVM and Random Forest algorithm improves the capacity of malware simulation to identify the program as benign or harmful. Their method has been extended to 600 applications. The experimental results indicate that their solution takes half-time and higher reminder rates than Androguard [105]. By utilizing the Random Forest machine learning method on picture characteristics created from APK samples, the suggested study could reach 84.14 percent detecting accuracy. Android malware has been detected daily such that malware analysts find it tough to identify it. For autonomous learning, they employed a neural network, and to be more specific, the detection is done and compared using three distinct classifiers: KNN, Random Forest and Decision Tree (DT).Less costly memory representation and hence speed up the learning process [106].

7. DISCUSSION AND COMPARISON

This section is focussing on browsing significant comparison among all previous works explained in section 6 which was the summary of related works in the field of malware detection in android operating system. The comparison illustrated the accuracy concept among different previous approaches towards malware detection as illustrated in Table 1. Also, the depended algorithms are compared clearly, in this table. So, different algorithms were depended by previous works (Intelligence Machine Learning, ROC, SVM, PPMDroid, and EveDroid).

Table 1. Comparison among of the related works

Ref.	Model	Methods/ Algorithm	Objective	Results
[3]	model that combines the advantages of static analysis, dynamic analysis	Intelligence Machine Learning	This study provides an effective and accurate solution to this problem	achieves high accuracy of malware detection via efficiency
[4]	Each malware detection technique's flaws were highlighted.		The objective of the paper is to create a user profiling method for the mobile identification of malware.	Based on mobile user profile, it may be utilized effectively.
[9]	Some classification algorithms have been evaluated in their research to assess best performance	f-method, receiver operating curve (ROC)	Their project was designed to test Android malware classification algorithms.	It was discovered that multi-layer perceptron Performs best with an accuracy of 99.4%.
[11]	Proposed a new classification method based on the findings of a longitudinal analysis	machine learning	The key lesson learned is that learning software development offers a promising way to identify malware over the long term	These results showed the potential of long-term malware identification approaches focused on evolution.
[74]	ONAMD Online Android Malware Detection Approach	the SVM and Random Forest algorithm		The experimental results indicate that their solution takes half-time and higher reminder rates than Androguard
[64]	removes Uncertainty permissions, and extracts certain permission features.	Metropolis algorithm	To learn and classify, use these essential permissions	Their solution decreases detecting, achieve 93.5 percent for the accuracy of harmful program detection.
[67]	Built FalDroid, a new framework that automatically classifies Android malware		Malware inspection and malware raising to avoid analysis.	Minimize malware investigation costs by picking representative samples only 8.5 % to 22 percent
[89]	The model also uses grouping techniques like stream	machine learning	It is essential for users to provide a fast and accurate detection method.	This study condenses the progression of malware detection techniques supported
[90]	First, the privacy problems of current static and dynamic malware detection methods are highlighted	PPMDroid method	To conserve bandwidth and speed up the process with many optimizations.	Large evaluation findings with real malware samples show the reliability and efficacy of their method
[91]	The method will combine both static and evolving effects of study	deep learning	The usage of cell phones is increasing and sadly, cyber criminals are constantly targeting mobile phones.	This reduced the reliability of those solutions
[92]	DL-Droid,	deep learning	To detect malicious apps from Android.	Their findings also emphasize the importance of enhanced input generation for complex analytical systems built

Ref.	Model	Methods/ Algorithm	Objective	Results
[95]	designed to allow detection of Android and IoT malware	EveDroid	Due to API changes, able to capture previously unseen activities.	The results also demon strate that EveDroid is more accurate and robust to malware evolvement compared to existing detection systems
[96]	Proposed the second part of CICAndMal2017	CICAndMal2017	the main targets for attackers to unleash destructive intentions	At the first layer, successful 95.3 % with Static-Based Malware Binary, 83.3 % with Dynamic-Based, and 59.7% with Dynamic-Based at the second layer.
[98]	The model used permissions to get an accuracy of approximately 80% and system calls to reach a classification accuracy of about 60%.	machine learning		The results suggested that permission data is better for malware detection than system call data.
[99]	It identifies Android malware from two static and dynamic analytical perspectives	machine learning classification	It is possible to effectively detect malware.	Improve precision and efficiency of detection.
[101]	study presents two detection approaches for end-to-end malware without human engineering	deep learning DexCNN and DexCRNN	In order to train the pre-processed sequences	DexCNN can achieve accuracy of 93.4 percent, while the DexCRNN can reach accuracy of 95.8 per cent.
[102]	Utilized to assess the method of detection and to validate the system's efficacy.	random forest algorithm	This method increases to a certain degree the detection accuracy	The sensitive allowance rules approach is used
[103]	DAMBA collects application's static and dynamic characteristics.	TANMAD Algorithm	to preserve the mobile client's limited resources	In order to improve efficiency and precision,
[104]	It is suggested that you conduct a Cloak and Dagger assault.	Cloak and Dagger attack algorithm	The market for mobile operating systems is expanding.	Enables the detection of all potentially harmful programs on a mobile device's operating system.
[105]	The detection will be measured using three distinct classifications (KNN), (RF) and (DT).	GIST descriptor	to create a malware application for Android devices	Achieve 84.14% accuracy detection
[106]	Their solution use the matrix representing the system calls gathered and the CNN model input	a neural network	For autonomous learning, employed a neural network, and to be more specific	Less costly memory representation and hence speed up the learning process

8. CONCLUSION

The article provides a foundational understanding of Android malware detection technologies. Malware detection is a cornerstone of the Android Operating System's security. A comparison was formed on the various techniques. The detection techniques that use hybrid analysis and use deep learning both of them are accurate and scalable as well as machine-based learning detection may discover previously undiscovered malware kinds, and may improve detecting efficiency performance. The strategy that reports all the constraints of static and dynamic analysis methodologies must be introduced to identify hybrid malware. Research is still ongoing in this area to enhance the accuracy and reliability of systems.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. Abdullah RM, Ameen SY, Ahmed DM, Kak SF, Yasin HM, Ibrahim IM, et al. Paralinguistic Speech Processing: An Overview, Asian Journal of Research in Computer Science. 2021;34-46.
2. Uma K, Blessie ES. Survey on android malware detection and protection using data mining algorithms," in 2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), 2018 2nd International Conference on. 2018;209-212.
3. Arshad S, Shah MA, Wahid A, Mehmood A, Song H, Yu H. Samadroid: a novel 3-level hybrid malware detection model for android operating system," IEEE Access. 2018;6: 4321-4339.
4. Amro B. Malware detection techniques for mobile devices, International Journal of Mobile Network Communications & Telematics (IJMNCT). 2017;7.
5. Ibrahim IM, Ameen SY, Yasin HM, Omar N, Kak SF, Rashid ZN, et al. Web Server Performance Improvement Using Dynamic Load Balancing Techniques: A Review, Asian Journal of Research in Computer Science. 2021;47-62.
6. Ahmed DM, Ameen SY, Omar N, Kak SF, Rashid ZN, Yasin HM, et al. A State of Art for Survey of Combined Iris and Fingerprint Recognition Systems," Asian Journal of Research in Computer Science. 2021;18-33.
7. Maulud DH, Ameen SY, Omar N, Kak SF, Rashid ZN, Yasin HM, et al. Review on Natural Language Processing Based on Different Techniques," Asian Journal of Research in Computer Science. 2021;1-17.
8. Bayazit EC, Sahingoz OK, Dogan B. Malware detection in Android systems with traditional machine learning models: a survey, in 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA). 2020;1-8.
9. Olorunshola OE, Oluyomi AO. ANDROID APPLICATIONS MALWARE DETECTION: A Comparative Analysis of some Classification Algorithms," in 2019 15th International Conference on Electronics, Computer and Computation (ICECCO). 2019;1-6.
10. Salih AA, Ameen SY, Zeebaree SR, Sadeeq MA, Kak SF, Omar N, et al. Deep Learning Approaches for Intrusion Detection," Asian Journal of Research in Computer Science. 2021;50-64.
11. Fu X, Cai H. On the deterioration of learning-based malware detectors for Android," in 2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion). 2019;272-273.
12. Alqahtani EJ, Zagrouba R, Almuhaideb A. A Survey on Android Malware Detection Techniques Using Machine Learning Algorithms," in 2019 Sixth International Conference on Software Defined Systems (SDS). 2019;110-117.
13. Ibrahim BR, Khalifa FM, Zeebaree SR, Othman NA, Alkhayyat A, Zebari RR, et al. Embedded System for Eye Blink Detection Using Machine Learning Technique," in 2021 1st Babylon International Conference on Information Technology and Science (BICITS). 2021;58-62.
14. Hassan RJ, Zeebaree SR, Ameen SY, Kak SF, Sadeeq MA, Ageed ZS, et al. State of art survey for iot effects on smart city technology: challenges, opportunities, and solutions," Asian Journal of Research in Computer Science. 2021;32-48.
15. Yahia HS, Zeebaree SR, Sadeeq MA, Salim NO, Kak SF, Adel AZ, et al. Comprehensive survey for cloud computing

based nature-inspired algorithms optimization scheduling, Asian Journal of Research in Computer Science. 2021;1-16.

16. Ageed ZS, Zeebaree SR, Sadeeq MM, Kak SF, Rashid ZN, Salih AA, et al. A survey of data mining implementation in smart city applications," Qubahan Academic Journal. 2021;1:91-99.
17. Ageed ZS, Zeebaree SR, Sadeeq MA, Abdulrazzaq MB, Salim BW, Salih AA, et al. A state of art survey for intelligent energy monitoring systems," Asian Journal of Research in Computer Science. 2021;46-61.
18. Abdullah DM, Ameen SY, Omar N, Salih AA, Ahmed DM, Kak SF, et al. Secure Data Transfer over Internet Using Image Steganography," Asian Journal of Research in Computer Science. 2021;33-52.
19. Kareem FQ, Ameen SY, Salih AA, Ahmed DM, Kak SF, Yasin HM, et al. SQL Injection Attacks Prevention System Technology, Asian Journal of Research in Computer Science. 2021;3-32.
20. Hasan DA, Zeebaree SR, Sadeeq MA, Shukur HM, Zebari RR, Alkhayyat AH. Machine Learning-based Diabetic Retinopathy Early Detection and Classification Systems-A Survey," in 2021 1st Babylon International Conference on Information Technology and Science (BICITS). 2021;16-21.
21. Ismael HR, Ameen SY, Kak SF, Yasin HM, Ibrahim IM, Ahmed AM, et al. Reliable communications for vehicular networks," Asian Journal of Research in Computer Science. 2021;33-49.
22. Abdulla AI, Abdulraheem AS, Salih AA, Sadeeq M, Ahmed AJ, Ferzor BM, et al. Internet of things and smart home security," Technol. Rep. Kansai Univ. 2020;62:2465-2476.
23. Abdulraheem AS, Salih AA, Abdulla AI, Sadeeq M, Salim N, Abdullah H, et al. Home automation system based on IoT; 2020.
24. Salih AA, Zeebaree S, Abdulraheem AS, Zebari RR, Sadeeq M, Ahmed OM. Evolution of mobile wireless communication to 5G revolution," Technology Reports of Kansai University. 2020;62: 2139-2151.
25. Dino HI, Zeebaree S, Salih AA, Zebari RR, Ageed ZS, Shukur HM, et al. Impact of Process Execution and Physical Memory-Spaces on OS Performance," Technology Reports of Kansai University. 2020;62:2391-2401.
26. Hamdi SJ, Ibrahim IM, Omar N, Ahmed OM, Rashid ZN, Ahmed AM, et al. A Comprehensive Study of Malware Detection in Android Operating Systems."
27. Ageed ZS, Ahmed AM, Omar N, Kak SF, Ibrahim IM, Yasin HM, et al. A State of Art Survey of Nano Technology: Implementation, Challenges, and Future Trends."
28. Jijo BT, Zeebaree SR, Zebari RR, Sadeeq MA, Sallow AB, Mohsin S, et al. A comprehensive survey of 5G mm-wave technology design challenges," Asian Journal of Research in Computer Science. 2021;1-20.
29. Abdulqadir HR, Zeebaree SR, Shukur HM, Sadeeq MM, Salim BW, Salih AA, et al. A study of moving from cloud computing to fog computing," Qubahan Academic Journal. 2021;1:60-70..
30. Yazdeen AA, Zeebaree SR, Sadeeq MM, Kak SF, Ahmed OM, Zebari RR. FPGA implementations for data encryption and decryption via concurrent and parallel computation: A review," Qubahan Academic Journal. 2021;1:8-16.
31. Ageed ZS, Zeebaree SR, Sadeeq MM, Kak SF, Yahia HS, Mahmood MR, et al. Comprehensive survey of big data mining approaches in cloud systems," Qubahan Academic Journal. 2021;1:29-38.
32. Abdulrahman LM, Zeebaree SR, Kak SF, Sadeeq MA, Adel AZ, Salim BW, et al. A state of art for smart gateways issues and modification, Asian Journal of Research in Computer Science. 2021;1-13.
33. AL-Zebari A, Zeebaree S, Jacksi K, Selamat A. ELMS–DPU ontology visualization with Protégé VOWL and Web VOWL, Journal of Advanced Research in Dynamic and Control Systems. 2019;11:478-85.
34. Zeebaree A, Adel A, Jacksi K, Selamat A. Designing an ontology of E-learning system for duhok polytechnic university using protégé OWL tool," J Adv Res Dyn Control Syst. 2019;11:24-37.
35. Kareem FQ, Zeebaree SR, Dino HI, Sadeeq MA, Rashid ZN, Hasan DA, et al. A survey of optical fiber communications: challenges and processing time influences, Asian Journal of Research in Computer Science. 2021;48-58.

36. Adel AZ, Zebari S, Jacksi K. Football Ontology Construction using Oriented Programming," Journal of Applied Science and Technology Trends. 2020;1:24-30.
37. SA AZ, Selamat A. Electronic Learning Management System Based on Semantic Web Technology: A Review, Int. J. Adv. Electron. Comput. Sci. 2017;4:1-6.
38. Abdullah SMSA, Ameen SYA, Sadeeq MA, Zeebaree S. Multimodal emotion recognition using deep learning," Journal of Applied Science and Technology Trends. 2021;2:52-58.
39. Abdullah RM, Abdulazeez AM, Al-Zebari A."Machine learning Algorithm of Intrusion Detection System," Asian Journal of Research in Computer Science. 2021;1-12.
40. Shukur H, Zeebaree SR, Ahmed AJ, Zebari RR, Ahmed O, Tahir BSA, et al. A state of art survey for concurrent computation and clustering of parallel computing for distributed systems," Journal of Applied Science and Technology Trends. 2020;1:148-154.
41. Tahir B, Ali Saktioto J, Fadhali M, Rahman R, Ahmed A. A study of FBG sensor and electrical strain gauge for strain measurements," Journal of optoelectronics and advanced materials. 2008;10:2564-2568.
42. Harki N, Ahmed A, Haji L. CPU scheduling techniques: A review on novel approaches strategy and performance assessment," Journal of Applied Science and Technology Trends. 2020;1:48-55.
43. Ahmed A, Ahmed O. Correlation pattern among morphological and biochemical traits in relation to tillering capacity in sugarcane (Saccharum Spp)," Acad J Plant Sci. 2012;5:119-122.
44. Ahmed AJ, Mohammed FH, Majedkan NA. An Evaluation Study of an E-Learning Course at the Duhok Polytechnic University: A Case Study," Journal of Cases on Information Technology (JCIT). 2022;24:1-11.
45. Ahmed O, Geraldes R, Ahmed A, DeLuca G, Palace J. Multiple sclerosis and the risk of venous thrombosis: a systematic review," in Multiple Sclerosis Journal. 2017;757-758.
46. Salim NO, Abdulazeez AM. Human diseases detection based on machine learning algorithms: A review," International Journal of Science and Business. 2021;5:102-113.
47. Salim NO, Zeebaree SR, Sadeeq MA, Radie A, Shukur HM, Rashid ZN. Study for Food Recognition System Using Deep Learning, in Journal of Physics: Conference Series. 2021; 012014.
48. Salim NO, Abdulazeez AM. Science and Business," International Journal. 2021;5: 102-113.
49. Eesa AS, Sadiq S, Hassan M, Orman Z. Rule generation based on modified cuttlefish algorithm for intrusion detection system," Uludağ University Journal of The Faculty of Engineering. 2021;26: 253-268.
50. Zebari DA, Haron H, Zeebaree SR, Zeebaree DQ. Multi-Level of DNA Encryption Technique Based on DNA Arithmetic and Biological Operations, in 2018 International Conference on Advanced Science and Engineering (ICOASE). 2018;312-317.
51. Eesa AS. Optimization Algorithms for Intrusion Detection System: A Review," International Journal of Research-Granthaalayah. 2020;8:217-225.
52. Sadeeq MA, Zeebaree S. Energy management for internet of things via distributed systems," Journal of Applied Science and Technology Trends. 2021;2:59-71.
53. Haji SH, Zeebaree SR, Saeed RH, Ameen SY, Shukur HM, Omar N, et al. Comparison of software defined networking with traditional networking," Asian Journal of Research in Computer Science. 2021;1-18.
54. Zebari S, Yaseen NO. Effects of parallel processing implementation on balanced load-division depending on distributed memory systems," J. Univ. Anbar Pure Sci. 2011;5:50-56.
55. Malallah H, Zeebaree SR, Zebari RR, Sadeeq MA, Ageed ZS, Ibrahim IM, et al. A comprehensive study of kernel (issues and concepts) in different operating systems," Asian Journal of Research in Computer Science. 2021;16-31.
56. Yasin HM, Zeebaree SR, Sadeeq MA, Ameen SY, Ibrahim IM, Zebari RR, et al. IoT and ICT based smart water management, monitoring and controlling system: A review," Asian Journal of Research in Computer Science. 2021;42-56.
57. Omer MA, Zeebaree SR, Sadeeq MA, Salim BW, Mohsin SX, Rashid ZN, et al. Efficiency of malware detection in android system: A survey," Asian Journal of

Research in Computer Science. 2021;59-69.

58. Zeebaree SR, Sallow AB, Hussan BK, Ali SM. Design and simulation of high-speed parallel/sequential simplified DES code breaking based on FPGA," in 2019 International Conference on Advanced Science and Engineering (ICOASE). 2019;76-81.
59. Zebari IM, Zeebaree SR, Yasin HM. Real time video streaming from multi-source using client-server for video distribution," in 2019 4th Scientific International Conference Najaf (SICN). 2019; 109-114.
60. Liu K, Xu S, Xu G, Zhang M, Sun D, Liu H. A review of android malware detection approaches based on machine learning," IEEE Access. 2020;8:124579-124607.
61. Sharma S, Krishna CR, Sahay SK. Detection of advanced malware by machine learning techniques," in Soft computing: Theories and applications, ed: Springer. 2019;333-342.
62. Ibrahim BR, Zeebaree SR, Hussan BK. Performance Measurement for Distributed Systems using 2TA and 3TA based on OPNET Principles," Science Journal of University of Zakho. 2019;7:65-69.
63. Maulud DH, Zeebaree SR, Jacksi K, Sadeeq MAM, Sharif KH. State of art for semantic analysis of natural language processing," Qubahan Academic Journal. 2021;1:21-28.
64. Yuan H, Sun W. Android Application Security Detection Method Based on Metropolis Algorithm," in 2019 IEEE 19th International Conference on Communication Technology (ICCT). 2019; 1280-1284.
65. Zeebaree S, Yasin HM. Arduino based remote controlling for home: power saving, security and protection," International Journal of Scientific & Engineering Research. 2014;5:266-272.
66. Zeebaree S, Zebari I. "Multilevel client/server peer-to-peer video broadcasting system, International Journal of Scientific & Engineering Research. 2014;5:260-265.
67. Fan M, Liu J, Luo X, Chen K, Tian Z, Zheng Q, et al. Android malware familial classification and representative sample selection via frequent subgraph analysis," IEEE Transactions on Information Forensics and Security. 2018;13:1890-1905.
68. Sadeeq MM, Abdulkareem NM, Zeebaree SR, Ahmed DM, Sami AS, Zebari RR. IoT and Cloud computing issues, challenges and opportunities: A review," Qubahan Academic Journal. 2021;1: 1-7, 2021.
69. Zeebaree S, Zebari RR, Jacksi K, Hasan DA. "Security approaches for integrated enterprise systems performance: A Review," Int. J. Sci. Technol. Res. 2019;8.
70. Hasan DA, Hussan BK, Zeebaree SR, Ahmed DM, Kareem OS, Sadeeq MA. The impact of test case generation methods on the software performance: A review," International Journal of Science and Business. 2021;5:33-44.
71. Jader OH, Zeebaree S, Zebari RR. A state of art survey for web server performance measurement and load balancing mechanisms," International Journal of Scientific & Technology Research. 2019;8:535-543.
72. Jacksi K, Ibrahim RK, Zeebaree SR, Zebari RR, Sadeeq MA. Clustering documents based on semantic similarity using HAC and K-mean algorithms," in 2020 International Conference on Advanced Science and Engineering (ICOASE). 2020;205-210.
73. Sadeeq MA, Abdulazeez AM. Neural networks architectures design, and applications: A review," in 2020 International Conference on Advanced Science and Engineering (ICOASE). 2020;199-204.
74. Riasat R, Sakeena M, Sadiq AH, Wang YJ. Onamd: An online android malware detection approach, in 2018 International Conference on Machine Learning and Cybernetics (ICMLC). 2018;190-196.
75. Ibrahim IM. Task scheduling algorithms in cloud computing: A review, Turkish Journal of Computer and Mathematics Education (TURCOMAT). 2021;12:1041-1053.
76. Ageed ZS, Ibrahim RK, Sadeeq M. Unified ontology implementation of cloud computing for distributed systems," Current Journal of Applied Science and Technology. 2020;82-97.
77. Zeebaree S, Ameen S, Sadeeq M. Social media networks security threats, risks and recommendation: A case study in the kurdistan region," International Journal of Innovation, Creativity and Change. 2020;13:349-365.
78. Sulaiman MA, Sadeeq M, Abdulraheem AS, Abdulla AI. Analyzation study for

gamification examination fields," Technol. Rep. Kansai Univ. 2020;62:2319-2328.
79. Yasin HM, Zeebaree SR, Zebari IM. Arduino based automatic irrigation system: Monitoring and SMS controlling," in 2019 4th Scientific International Conference Najaf (SICN). 2019;109-114.
80. Sadeeq M, Abdulla AI, Abdulraheem AS, Ageed ZS. Impact of electronic commerce on enterprise business," Technol. Rep. Kansai Univ. 2020;62:2365-2378.
81. Alzakholi O, Shukur H, Zebari R, Abas S, Sadeeq M. Comparison among cloud technologies and cloud performance," Journal of Applied Science and Technology Trends. 2020;1:40-47.
82. Zeebaree S, Zebari RR, Jacksi K. Performance analysis of IIS10. 0 and Apache2 Cluster-based Web Servers under SYN DDoS Attack," TEST Engineering & Management. 2020;83:5854-5863.
83. Ageed Z, Mahmood MR, Sadeeq M, Abdulrazzaq MB, Dino H. Cloud computing resources impacts on heavy-load parallel processing approaches," IOSR Journal of Computer Engineering (IOSR-JCE). 2020;22:30-41.
84. Sallow A, Zeebaree S, Zebari R, Mahmood M, Abdulrazzaq M, Sadeeq M. Vaccine tracker, SMS reminder system: Design and implementation; 2020.
85. Sadeeq MA, Zeebaree SR, Qashi R, Ahmed SH, Jacksi K. Internet of Things security: A survey," in 2018 International Conference on Advanced Science and Engineering (ICOASE). 2018;162-166.
86. Abdulazeez AM, Zeebaree SR, Sadeeq MA. Design and implementation of electronic student affairs system," Academic Journal of Nawroz University. 2018;7:66-73.
87. Sallow AB, Sadeeq M, Zebari RR, Abdulrazzaq MB, Mahmood MR, Shukur HM, et al. An investigation for mobile malware behavioral and detection techniques based on android platform," IOSR Journal of Computer Engineering (IOSR-JCE). 2020;22:14-20.
88. Dino H, Abdulrazzaq MB, Zeebaree S, Sallow AB, Zebari RR, Shukur HM, et al. Facial expression recognition based on hybrid feature extraction techniques with different classifiers," TEST Engineering & Management. 2020;83:22319-22329.
89. Murtaz M, Azwar H, Ali SB, Rehman S. A framework for Android Malware detection and classification," in 2018 IEEE 5th International Conference on Engineering Technologies and Applied Sciences (ICETAS). 2018;1-5.
90. Cui H, Zhou Y, Wang C, Li Q, Ren K. Towards privacy-preserving malware detection systems for android," in 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS). 2018;545-552.
91. Shibija K, Joseph RV. A Machine Learning Approach to the Detection and Analysis of Android Malicious Apps," in 2018 International Conference on Computer Communication and Informatics (ICCCI). 2018;1-4.
92. Alzaylaee MK, Yerima SY, Sezer S. DL-Droid: Deep learning based android malware detection using real devices," Computers & Security. 2020;89:101663.
93. Wang Z, Liu Q, Chi Y. Review of android malware detection based on deep learning," IEEE Access. 2020;8:181102-181126.
94. Samra AAA, Qunoo HN, Al-Rubaie F, El-Talli H. A survey of static android malware detection techniques," in 2019 IEEE 7Th palestinian international conference on electrical and computer engineering (PICECE). 2019;1-6.
95. Lei T, Qin Z, Wang Z, Li Q, Ye D. EveDroid: Event-aware Android malware detection against model degrading for IoT devices," IEEE Internet of Things Journal. 2019;6:6668-6680.
96. Taheri L, Kadir AFA, Lashkari AH. Extensible android malware detection and family classification using network-flows and API-calls," in 2019 International Carnahan Conference on Security Technology (ICCST). 2019;1-8.
97. Patel ZD. Malware detection in android operating system," in 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN). 2018;366-370.
98. Leeds M, Keffeler M, Atkison T. A comparison of features for android malware detection," in Proceedings of the SouthEast Conference. 2017;63-68.
99. Qing-Fei W, Xiang F. Android Malware Detection Based on Machine Learning," in 2018 4th Annual International Conference on Network and Information Systems for Computers (ICNISC). 2018;434-436.
100. Sabhadiya S, Barad J, Gheewala J. Android malware detection using deep learning," in 2019 3rd International Conference on

Trends in Electronics and Informatics (ICOEI). 2019;1254-1260.

101. Ren Z, Wu H, Ning Q, Hussain I, Chen B. End-to-end malware detection for android IoT devices using deep learning," Ad Hoc Networks. 2020;101:102098.
102. Lu T, Hou S. A Two-Layered Malware Detection Model Based on Permission for Android," in 2018 IEEE International Conference on Computer and Communication Engineering Technology (CCET). 2018;239-243.
103. Zhang W, Wang H, He H, Liu P. DAMBA: detecting android malware by ORGB analysis," IEEE Transactions on Reliability. 2020;69:55-69.
104. Omelchenko T, Nikishova A, Umnitsyn M, Sadovnikova N, Parygin D, Kostyukov A. Protection Software for Mobile Operating Systems," in 2018 International Conference on System Modeling & Advancement in Research Trends (SMART). 2018;54-59.
105. Darus FM, Salleh NAA, Ariffin AFM. Android malware detection using machine learning on image patterns," in 2018 Cyber Resilience Conference (CRC). 2018;1-2.
106. Agrawal P, Trivedi B. A survey on android malware and their detection techniques," in 2019 IEEE International conference on electrical, computer and communication technologies (ICECCT). 2019;1-6.

Peer-review history:
The peer review history for this paper can be accessed here:
https://www.sdiarticle4.com/review-history/73749