An Evaluate on Ensemble Learning Approach for an Effective Android Malware Detection

[1]SUMALATHA POTTETI
Assistant Professor, Department of Computer Science and Engineering, Bhoj Reddy Engineering College for Women,Hyderabad,Telangana, India.

[2]Dr. G. S. MAHALAKSHMI
Associate Professor, Department of Computer Science and Engineering, Anna University, Chennai, Tamilnadu, India.

Abstract:

As the number of Android-based devices grows in popularity throughout the world, so do thevulnerable to various attacks. The base installation package of Android is Android Package (APK). Lots of vulnerabilities are found in APK files in terms of malware and other assaults. The main focus is on predicting the malware in android APK files in order to better predict assaults with assorted vulnerabilities. To accomplish multi-dimensional outcomes, the research work incorporates the use of numerous methodologies in machine learning. With the growing number of smartphone apps and the widespread use of Android by mobile users, security concerns are becoming increasingly essential. These concerns must be addressed so that vulnerabilities can be avoided and recognised in advance. Users of smartphone apps can be linked to a warning about specific vulnerabilities using this method. Users of mobile devices can quickly install APK files from many sources without experiencing any negative consequences. A mechanism and algorithm for predicting harmful code in Android APKs must be developed and implemented. The work integrates the Android APK datasets for analytics using ensemble learning. There are two types of APKs: benign and malignant. The goal of this study is to extract deep signatures from these APKs so that a training dataset may be created. A number of APK files are analysed, with benign and the other being malignant. Then there's the checking of permissions in each APK and their consequences. Following that, the production of a cleaned dataset is planned in order to train the model for prediction. Then, in order to predict, any other random APK that isn't present in those APKs is added to the predictive analytics. Then there's a prognosis of the likelihood of having harmful key points in the new APK under investigation. The entire prediction and performance metrics are evaluated using machine learning based methodologies on a variety of parameters such as time, cost, accuracy.

Keywords: MachineLearning,Random Forest Approach, Support Vector Classifier, Regression and Ensemble Learning Based approaches.

DOI Number: 10.14704/nq.2022.20.13.NQ88094 **Neuro Quantology 2022; 20(13):722-736**

1. Introduction:

In this age of information technology and high-speed data transport, computer forensic specialists' major concern is protection and safety. Various tactics and technology are accessible in this region that can deliver malware or truly damaging packets through the network, allowing virtual attacks to damage the data channel. Malware is usually divided into two categories: malware and non-malicious applications, based on a variety of factors such as reproduction trending and technique, development, dissemination process, and containment technique.

Malware is software that is specifically designed to harm a computer, a server client, or a computer network. (Instead, software that causes unintentional harm as a result of flaws is referred to as a software bug.) Malware includes computer viruses, worms, Trojan horses, ransomware, spyware, and adware, among other things. There is a wide range of malware software that necessitates meticulous attention to the predictions.

Ransomware is a term for programmes that operate unlawfully against the wishes of computer users. For example, Sony music discs Compact inadvertently installed a rootkit on purchasers' machines to prevent unauthorised copying, but it also recorded users' listening patterns and caused additional security issues.

To defend against malware proliferation, prevention, and recovery from harmful activities and attacks associated with malware, a variety of antivirus applications, firewalls, and other techniques are used.

At the moment, black hat hackers and the government are using malware to steal personal, financial, and business information. Many infectious programmes, including the first Internet Worms, were written as a test or prank against security systems.

Malware is used against government and business websites to capture information that is protected or to disrupt its activity in general. Malware, on the other hand, can be used against individuals to collect information such as personal identifying numbers or data, bank or credit card numbers, and passwords.

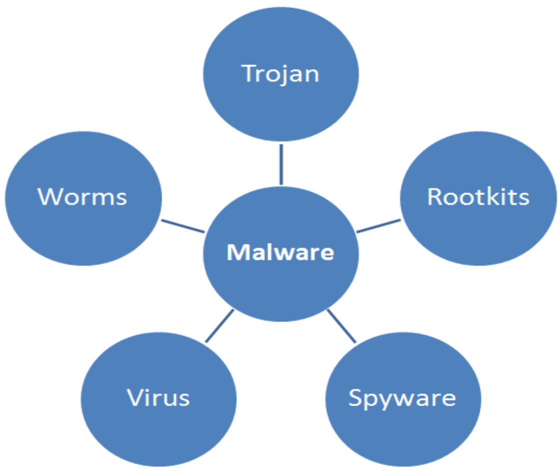


Figure 1.1 Malware Taxonomy

2. LiteratureSurvey:

Malicious malware has become more prevalent as broadband internet connectivity has expanded. Since 2003, the bulk of regularly disseminated viruses and worms have been developed to control users' computers for unlawful objectives, such as sending e-mail, hosting contraband files like child pornography, or conducting widespread negation attacks as a technique of externalisation.

Spyware is a term for automated software that track users' online activities, display unwanted advertisements, or divert affiliate marketing earnings. Spyware applications are often designed for security troubleshooting and do not spread like malware. The Sony BMG rootkit was designed to prevent unauthorised copying; instead, customers' listening habits were frequently recorded, resulting in an additional security issue. It could potentially be hidden and bundled with unrelated user-installed software.

Ransomware has an effect on a hacked operating system, allowing payment to restore it to its previous state. In ransomware, there are two variables: ransomware cryptography and ransomware locker. The ransomware locker simply locks down a computer device that is missing its contents. Unlike traditional ransomware, which locks down and encrypts the contents of a device, programmes like CryptoLocker, for example, encrypt files safely and then decode them after paying a large sum of money. Some ransomware has been used to make money by clicking on the scam, indicating that the owner of your laptop has paid for it by clicking on an internet advertisement connection. In 2012, it was estimated that 60 to 70% of all active viruses employed some type of click fraud.

Sabotaging software, primarily for political motives, can be used in addition to making unlawful funds. Stuxnet, for example, was created to harm extremely specific industrial devices. Politically motivated attacks on Sony Pictures Entertainment (25 November 2014, using malware known as Shamoon or W32. Disttrack) and Savia Aramco (August 2012) spread across and shut down large computer networks, including massive file deletions and corruptions of master boot logs described as a "computer killing."

Instead of taking action, the most well-known types of ransomware, viruses, and worms are well-known for the way they spread. A machine virus is a software that instals itself on a target device without the user's knowledge or consent, and once installed, the virus spreads to other executables and applications (including the operating system itself). A worm, on the other hand, is a standalone malware application that actively communicates to other computers across a network. These definitions show that a virus allows a user to install an infected viral programme or operating system, whereas a worm spreads.

Software that is frequently hidden inside an apparently safe application can make and insert copies of itself as well as other potentially hazardous programmes or files (such as data destruction) Invasion of PE, for example, is a widely used approach for spreading malware and inserting extra data or executable code into PE files.

"Lock-Screens" or screen lockers are ransomware from a "cyber police" that stops screens on Windows or Android devices with an allegation error when it comes to gathering unlawful material, in an attempt to scare victims into paying a ransom. Jisut and SLocker have a greater impact on Android devices than other screen lockers.

A Trojan horse is a deceitful strategy that masquerades as a typical, harmless scheme or service in order to trick a victim into installing it. A Trojan horse usually contains a hidden disruptive feature that you can enable when you run the programme. This is based on the Trojan horse, which was used by thieving Trojans to take the city of Troy in Ancient Greece.

In general, Trojan horses are disseminated by social engineering techniques, such as sending a null e-mail attachment (e. g., a normal form to fill up) or downloading it from a driver. Even if they are capable of handling something, certain modern methods are used as backdoors to contact the controller (by calling the home), who can then gain unauthorised access to the device and likely install additional tools, such as a keylogger to steal sensitive information, security software, or adware to generate revenue for the Trojan operator. Cryptographers can limit the use of resources in order to avoid discovery and/or only operate during periods of inactivity.Trojans, unlike computer viruses and worms, do not attempt to inject or spread to another file.

Mac users who were eligible to get passwords from a variety of channels, such as auto-fill browsers, the Mac-OS keychain, and the password vault, were hit by the newest Proton Remote Access Trojan (RAT) in spring 2017.

To prevent detection, malicious software must be installed on a device. The programme packages known as rootkits enable this camouflage by modifying the operating system of the host to disguise a customer. RootKits protect the readability of a device's data by avoiding the exposure of a harmful procedure in the device process list.

Some harmful programme routines do not truly hide themselves; instead, they defy detection and/or removal. The Jargon File storey about two programmes infecting a Xerox CP-V time sharing system shows previous examples of this behaviour:

Each ghost job will recognise that the other worker has died, and a new duplicate of the just stopped programme will start in a matter of milliseconds. The only way to destroy the two fantasies was to kill them both at the same time (which was extremely tough) or purposely crash the machine.

Backdoors are a way to get around normal security procedures, usually via using a network connection such as the Internet. Once a device has been compromised, one or more backdoors can be installed to allow the consumer to gain access to it in the future.

The theory has long been suggested, but it has never been proven. Backdoors are built into products by manufacturers in order to give users with technological assistance. In 2014, US government agencies allegedly deported equipment they purchased to underground workshops, where the department installed software or gadgets that allow remote access to networks, and is acknowledged to be one of the most efficient operations in the world.

Since early 2015, a wide range of malware has employed a number of techniques to avoid detection and analysis. From the most well-known to the less well-known:

1. Avoidance and recognition of the world's fingerprints.
2. Techniques for identifying confounding automated technologies. This prevents ransomware by altering the registry, which malware uses to track technologies like signature-based antivirus software.
3. Timing evasion. This is due to the malware's ability to run at specific periods or in

response to specific user actions, such as during the boot process, when the bulk of the time it is still sleeping.

4. Internal data in order to evade virus detection by automatic software.

Adware, which uses stolen certificates to disable anti-malware and virus protection, is a more widespread tactic in 2015; the adware comes with technical cures.

Today, one of the sophist and stealthiest kinds of evasion is the use of hidden technologies, specifically stegomalware. Cabaj*et al.* conducted a stegomalware survey in 2018.

Another type of evasion strategy is fileless malware or sophisticated volatile threats (AVTs). Malware without archives does not require the execution of a file. It runs in memory and executes wrongdoings with current machine resources. There are no executable antivirus files or testing techniques to analyse because there is no device registry, making it very impossible to detect malware. Only by catching fileless malware in real time can it be recognised. These types of attacks have recently gained traction, with a 432 percent increase in 2017 and a 35 percent increase in 2018. Such attacks are difficult to carry out, but they are becoming increasingly widespread thanks to exploit kits.

- The term "system" in this context and in general can refer to anything from a single programme to a large network, as well as a whole device and operating system.
- A device's vulnerability to malware is influenced by a number of factors:

Malware exploits security flaws in operating system architecture, software (such as the prior edition of Windows XP-assisted Microsoft Internet Explorer), and insecure versions of browser plugins (such as Adobe Flash Player, Adobe Acrobat and Editor, or Java SE) (security glitches or vulnerabilities). It will also install updated versions of those plugins. Protection alerts from plug-in providers reveal changes in security. CVE IDs are assigned to various vulnerabilities and are found in the national vulnerability databases of the United States. Secunia PSI is a free-to-use tool that scans a computer and tries to update any vulnerable or out-of-date software.

Malware authors seek to take advantage of flaws or blind spots. A common strategy is to exploit the flaw of a buffer overrun, which occurs when data storage software does not prevent the delivery of more data than the buffer can handle. Malware can deliver buffer overflowing data, executable harmful coding, or post-end data; when accessed, it acts according to the attacker's wishes, not the legitimate program's.

According to Symantec's 2018 Internet Protection Threats Survey, the number of malware variants increased to 669,947,865 in 2017, more than doubling the number of malware variants in 2016. (ISTR).

The operating system was normally run on the built-in hard drives, but if they had additional booters, such as a disc, CD-ROM, DVD-ROM, USB flash drive, or network, they may be booted from a different boot unit. When one of these computers was available, the machine was frequently set to boot from it. Normally, there would be none, and the consumer would have to specifically boot the device by inserting a CD into the optical drive, for example, to download an operating system. Computers can be designed to run software on usable media even if it is not booting up, for example, to run a CD or USB drive automatically.

Malware distributors deceive users into booting or working from a compromised machine or medium. A virus, for example, may add authoritative code to every USB stick inserted onto an infected PC. Those who connect the stick to another USB device will be compromised and infected in the same way. Any device that plugs into a USB port, such as lighting, fans, speakers, toys, or even optical microscopes, could be exploited to spread malware. In the event of insufficient quality control, devices may become contaminated during manufacture or distribution.

By infecting a machine, you can block it from booting from your internal hard drive rather than from hardware. While such keys are pressed during booting, deliberate booting on another system is still possible.

Previously, email programmes would open the JavaScript code containing HTML email right away. Users will also send emails with hidden attachments. According to CSO Online, the main malware distribution tool is emails, which account for 92 percent of virus shipments worldwide, according to Verizon's 2018 Data Breach Inquiry Study.

The number of times a machine can be altered by a consumer or a computer programme is determined by privileges. In fraudulently created computer systems, both people and machines may have greater rights than they should, and malware can take advantage of this. Overprivileged users and overprivileged code are the two strategies used by ransomware.

Some programmes provide all users the ability to change their internal structures, and these users are now considered privileged. On a daily basis, there was little distinction between a manager or root and a device user in this basic operating style for early microcomputers and home computer systems. Non-administrator users of architecture in other applications have too much power, because internal configurations can be altered. In some contexts, users have too much power because administrator or similar status has been assigned to them incorrectly.

Some programmes, known as overprivileged code, allow users to control all of the user's privileges. The basic concept was also used for early microcomputers and home computing systems. Malware that runs as overprivileged code can leverage this privilege to disturb the device. Most common operating systems, as well as many application-scriptions, allow code to do a wide range of functions, with the device typically allowing the user to exercise all of his or her privileges while running code. As a result, consumers are vulnerable to ransomware in the context of an e-mail attachment, whether disguised or not. Homogeneity can be a flaw. For example, if every machine in a network uses the same operating system, a single worm can take advantage of all of them. Microsoft Windows and Mac OS X, in particular, have a market share that allows them to subvert a large number of systems by exploiting a flaw in either operating system. Short-term expenditures for preparation and repair may increase as a result of diversity, such as adding linux machines to boost resilience. Until all nodes are authorised by the same directory provider, a few isolated nodes will keep the networks up and running, allowing compromised nodes to recover. The cost of a complete shutdown might be used to eliminate such a distinct, practical redundancy at the expense of increasing sophistication and decreasing usability in the form of single sign-on authentication.

With malware attacks becoming more common, the focus is shifting away from viruses and spyware and toward malware security and tools designed specifically to combat malware. (For other protection and recovery actions, such as backup and recovery strategies, see the Computer Virus report.) Rebooting software is frequently beneficial in preventing ransomware by undoing malicious changes.

The paper presents a unique mechanism for protecting the web and operating system against malware while also ensuring greater and more efficient proof and virus test rates. This study is one-of-a-kind and represents outstanding work in the field of web-based malware and OS-based malware defence.

The inscription describes how to classify and analyse various technological software using both static and dynamic approaches in a positive and beneficial way.

According to Wang, W. *et al.*, 2019, Android security incidents have been more common in recent years. To improve the accuracy and efficiency of large-scale Android malware detection, the authors suggest a hybrid model based on deep autoencoder (DAE) and convolutional neural network (CNN). To improve malware detection accuracy, we first review the high dimensional properties of Android apps and employ numerous CNNs to identify Android malware. In the serial convolutional neural network design, we use Relu as a non-linear function to improve the sparseness and 'dropout' role (CNN-S). The convolutionary layer and the pooling layer work in tandem with the full link layer to improve functionality extraction. Under these conditions, CNN-S displays good function extraction and malware detection capabilities. Second, to reduce training time, we deploy a deep self-encoder as a CNN pre-training method. With deep autoencoder and CNN templates, the mix allows you to learn more diverse trends (DAE-CNN). We run testing on 10,000 benign and 13,000 dangerous apps. CNN-S outperforms traditional Android malware detection techniques by a wide margin. Although the DAE-CNN model's training period is 83 percent less than the CNN-S model's, the CNN-S model's precision is raised by 5% when compared to

SVM.

The identification of safety issues, which Albasir, A. *et al.*, 2020 focused on, is essentially a key difficulty for wireless applications. Malicious software (apps) can be identified, downloaded, and activated in online markets, according to several recent incidents (eg Google Play Store). These malicious programmes can collect and send personal information from millions of users to servers. In this paper, we propose a method for making wireless devices' power utilisation more resistant to rogue applications. To extract vector features based on AE's reconstruction errors, two stages are taken: The stacks Restrained Boltzmann AutoEncoders (AE) and Principal Component Analytics (PCA) are used in Feather Extraction (PCA). (ii) the classification classifier, which is used to train the single-class vector help for the classification task. A genuine collection of measurements is used to validate the technique. The results demonstrate that the reconstruction error of AEs can be viewed as a good discriminating, indicating a favourable capacity. The detection accuracy attained surpasses that of the previously reported technique, reaching up to 98 percent in some cases.

Wireless Sensor Networks (WSNs) comprise a variety of sensor devices that shape a network without a predetermined design and are put in a specific region, according to Batista, F. K. *et al.*, 2020. Malware has recently grown in popularity as a potential weakness for the Internet of Things, and hence for these networks. Malware dissemination on wireless sensor networks has been studied from many perspectives in numerous of the presented models, disregarding individual functionality. The primary goal of this research is to describe an agent-based model for analysing malware that is transmitted across these networks, as well as its agents, coefficients, and transitional rules. Finally, specific sample simulations are employed.

According to Pan, Y. *et al.*, 2020, the widespread use of the Android operating system has resulted in an increase in Android malware in recent years. Users are exposed to serious threats, including personal data leaking and sophisticated theft, as a result of the installation and application of Android malware on devices without their knowledge or authorization. Several strategies have been proposed by researchers and physicians to combat these dangers. Statistical analysis is one such tool that is often used to discover malware on Android software and may quickly detect malware. From January 2014 to March 2020, the work will conduct a systematic literature evaluation with 98 publications in order to present an understandable summary of the most recent work on Android malware detection with static study. The authors divided state analysis into four sections for Android malware identification based on device functionality, namely the Android system for characteristic reasons, the opcode method, software based on the graph, and symbolic execution technique. The authors then use static research to analyse the malware detection feature, as well as empirical data to examine the efficacy of various models in Android malware recognition. Finally, static analysis has been found to be effective in detecting malware on Android devices. In the non-neural network model, the tentative conclusion is that the neural network method outperforms the Android malware identification model. Static research, on the other hand, has a number of drawbacks. As a result, based on the current research community, it is suitable to rely on some new strategies to improve Android malware detection. Furthermore, a single platform is necessary for evaluating the efficacy of a range of malware detection approaches in Android.

According to Liu, K. *et al.*, 2020, Android apps are rapidly evolving throughout the smartphone ecosystem. On the topic of Android Malware detection, several experts researched and improved concepts and methodologies from diverse perspectives. Machine learning, according to existing research, is the most efficient and promising method for detecting Android malware. However, there are papers devoted to machine learning that explore many issues relating to Android malware detection. We believe that our research adds to previous research by covering a wider range of issues. This paper presents a comprehensive overview of malware detection algorithms for Android that are based on machine learning. The study will present a brief history of Android software, covering the creation of Android devices, security protocols, and malware classification for Android. The paper then evaluates and summarises the current status of research from a variety of perspectives, including sample collection, data preprocessing, function selection, machine learning models, algorithms, and detector performance evaluation. Finally, we look at the possibilities for future machine learning malware detection research on Android. This research provides researchers with a comprehensive picture of Android machine learning virus detection. Subsequent study may lead to the commencement of new studies and the direction of research in the field as a whole.

3. Android Package (APK):

APK stands for Android Package or Android Package Kit. It is the file format that Android uses to distribute and install apps.

An APK contains all the elements that an app needs to install correctly on your device. An APK is an archive file that contains multiple files and some metadata about them. APKs are a variant of the JAR (Java Archive) file format, since a lot of Android is built in Java. APK files allow you to install apps on your Android phone, they are like the APPX files used to install store apps on Windows 10.

i) Different types of malwares will affect mobile:

1. ***Downloading malicious Apps***: The most common method hackers use to spread malware is through apps and downloads.The apps you get an official app store are usually safe, but apps that are pirated or come from less legitimate sources often also contain malware. These are apps that appear to be legitimate, but instead contain spyware or other type of malware.
2. ***Using a mobile device with operating system vulnerabilities:***The mobile device itself may have vulnerabilities that hackers can exploit. These vulnerabilities are discovered quickly and patched up, if you are not regularly updating the software on your phone, your device will be vulnerable.
3. ***Opening suspicious emails:*** Most of the people will receive an email that says you have won something. You open the email and click on then link, and nothing happens, or you have been taken to a dummy site. But malware was downloaded and installed on your phone. The data on your phone may now be exposed to that hacker. The solution for this malware is trying to avoid opening suspicious emails on your mobile.
4. ***Using Non-secure Wi-Fi/ URLs:***If you are accessing insecure websites, you run the risk of exposing sensitive data transmitted from your device. You are also more susceptible to man-in-the-middle attacks and being exposed to malware. So, avoid using insecure websites and Wi-Fi networks and consider using antivirus protection and a VPN on your phone to secure Wi-Fi communication.
5. ***Receiving text message/ voicemail phishing:***A text message or a voicemail from what appears to be a legitimate source asking for personal information either about you or your device. Hackers often use this information to steal whatever data they can, including social security numbers, credit card data etc. They may even be able to use it to make a targeted attack to install malware on your mobile. Whenever you get a text like this, call the company on their legitimate mobile and verify with them. Never give out sensitive information through a text.

ii) How to protect your mobile device from malwares:

1. ***Use a VPN:*** A Virtual private network is a secure tunnel that lets you access and share information securely over public Wi-Fi networks.
2. ***Download apps only from reputable sources***: Unofficial app stores are more likely to be sources of malware-infected apps.
3. ***Do mobile vulnerability scanning:*** Use a vulnerability scanner like security metrics mobile for your mobile device.
4. ***Update software and hardware:*** Companies often release updates on mobile devices that address potential vulnerabilities.
5. ***Have mobile device policies in place:*** whether your company owns the devices, or your employees use their own, you need to have security policies set up that address the use of mobile devices.

4. Machine Learning to Detect Android Malware:

Malware detection in Android can be performed in two ways; signature-based detection methods and behaviour-based detection methods. The signature-based detection method is simple, efficient, and produces low false positives. The binary code of the application is compared with the signatures using a known malware database. However, there is no possibility to detect unknown malware using this method. Therefore, the behaviour-based/anomaly-based detection method is the most commonly used way. This method usually borrows techniques from machine learning and data science. Many research studies have been conducted to detect Android malware using traditional ML-based methods such as Decision Trees (DT) and Support Vector Machines (SVM) and novel DL-based models such as Deep Convolutional Neural Network (Deep-CNN) [40] and Generative adversarial networks [41]. These studies have shown that ML can be effectively utilised for malware detection in Android [9]. Most of these studies used datasets such as Drebin [42], Google Play [43], AndroZoo [44], AppChina [45], Tencent [46], YingYongBao [47], Contagio [48], Genome/MalGenome [49], VirusShare [50], IntelSecurity/MacAfee [51], MassVet [52], Android Malware Dataset (AMD) [53], APKPure [54], Anrdoid Permission Dataset [55], Andrototal [56], Wandoujia [57], Kaggle [58], CICMaldroid [59], AZ [60], and Github [61] to perform experiments and model training in their studies.

Hybrid Analysis with Machine Learning:

It is a approach which can be used in ML based Android malware detection. ML algorithms such as DT,J48,RF,KNN,KMean and SVM can be applied to all these approaches.

Different types of malware detection approaches:

1. ***Signature-based malware detection:*** it was detected using ML algorithms after the feature extraction process. After the feature extraction, sensitive API calls were also analysed before applying ML algorithms. Documents were collected such as reviews, user documents and app descriptions before following a similar approach as the signature-based method. It was identified that the behavioural based approach is better than the signature-based approach. If the topic modelling is combined with that approach, it was possible to achieve good results.
2. ***Hybrid Analysis Method***: it is created when the dynamic analysis method is integrated with the static analysis method. According to this study, the SVM classifier with the hybrid analysis method performed better than the other ML algorithms.

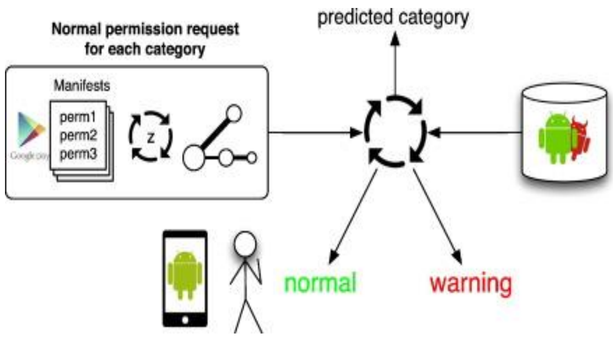


Figure 4.1 APK Analysis

5. ProposalProcess:

As the number of Android-based devices grows in popularity throughout the world, so do the vulnerabilities, because users are installing apps from a wider variety of sources. There must be a paradigm and technique presented so that APK's flaws may be recognised in advance.

Ensemble learning is the process of combining different models, such as classifiers or experts, to tackle a specific computer intelligence issue. Ensemble learning is mostly used to enhance the performance of a model (classification, prediction, function approximation, etc.) or lower the risk of an unlucky selection of a poor one. Ensemble learning. Aside from establishing a confidence level to the model's choice, ensemble learning may also be used to choose optimum (or near-optimal) features and to combine data from many sources. This article focuses on classification-related applications of ensemble learning, however the principles presented below may be readily applied to function approximation or prediction type issues.

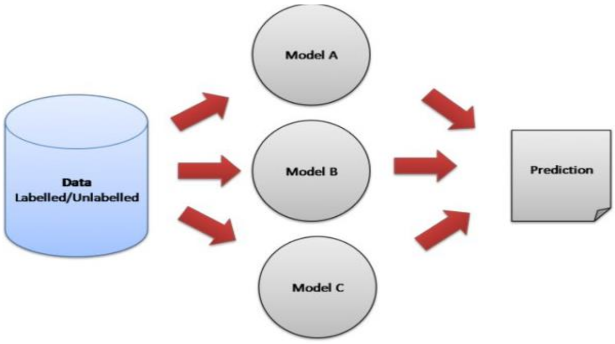


Figure 5.1 Ensemble Based Hybrid Approach

An ensemble-based system is obtained by combining diverse models (henceforth classifiers). Therefore, such systems are also known as multiple classifier systems, or just ensemble systems. There are several scenarios where using an ensemble based system makes statistical sense, which are discussed below in detail. However, in order to fully and practically appreciate the importance of using multiple classifier systems, it is perhaps instructive to look at a psychological backdrop to this otherwise statistically sound argument: we use such an approach routinely in our daily lives by asking the opinions of several experts before making a decision. For example, we typically ask the opinions of several doctors before agreeing to a medical procedure, we read user reviews before purchasing an item (particularly big ticket items), we evaluate future employees by checking their references, etc. In fact, even this article is reviewed by several experts before being accepted for publication. In each case, a final decision is made by combining the individual decisions of several experts. In doing so, the primary goal is to minimize the unfortunate selection of an unnecessary medical procedure, a poor product, an unqualified employee or even a poorly written and misguiding article.

Model Selection: This is perhaps the primary reason why ensemble based systems are used in practice: what is the most appropriate classifier for a given classification problem? This question can be interpreted in two different ways: i) what type of classifier should be chosen among many competing models, such as multilayer perceptron (MLP), support vector machines (SVM), decision trees, naive Bayes classifier, etc; ii) given a particular classification algorithm, which realization of this algorithm should be chosen - for example, different initializations of MLPs can give rise to different decision boundaries, even if all other parameters are kept constant.

The most commonly used procedure - choosing the classifiers with the smallest error on training data - is unfortunately a flawed one. Performance on a training dataset - even when computed using a cross-validation approach - can be misleading in terms of the classification performance on the previously unseen data. Then, of all (possibly infinite) classifiers that may all have the same training - or even the same (pseudo) generalization performance as computed on the validation data (part of the training data left unused for evaluating the classifier performance) - which one should be chosen? Everything else being equal, one may be tempted to choose at random, but with that decision comes the risk of choosing a particularly poor model. Using an ensemble of such models - instead of choosing just one - and combining their outputs by - for example, simply averaging them - can reduce the risk of an unfortunate selection of a particularly poorly performing classifier. It is important to emphasize that there is no guarantee that the combination of multiple classifiers will always perform better than the best individual classifier in the ensemble.

Nor an improvement on the ensemble's average performance can be guaranteed except for certain special cases (Fumera 2005). Hence combining classifiers may not necessarily beat the performance of the best classifier in the ensemble, but it certainly reduces the overall risk of making a particularly poor selection.

In order for this process to be effective, the individual experts must exhibit some level of diversity among themselves, as described later in this article in more detail. Within the classification context, then, the diversity in the classifiers – typically achieved by using different training parameters for each classifier – allows individual classifiers to generate different decision boundaries. If proper diversity is achieved, a different error is made by each classifier, strategic combination of which can then reduce the total error.

Neuro Quantology | October 2022 | Volume 20 | Issue 13 | Page 722-736 | doi: 10.14704/nq.2022.20.9.NQ88094

6. Ensemble Learning:

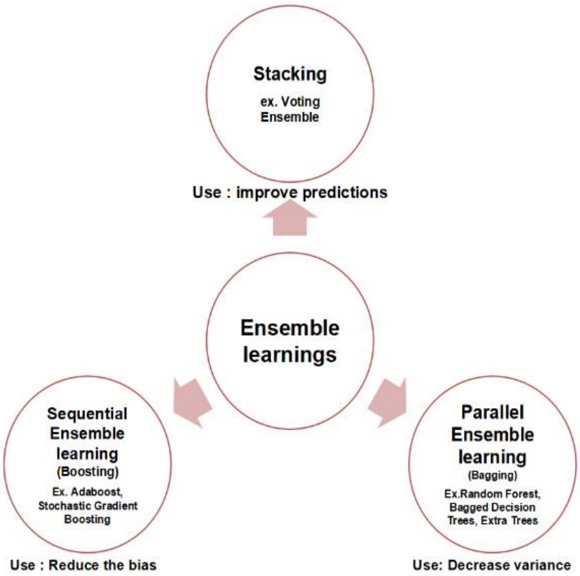


Figure 6.1 Ensemble Learning Based Approaches

One use of this statistical technique is data generation from a polynomial equation, followed by the prediction of each variable's value. Polynomial regression is based on the theory of the generic linear model, which serves as a foundation. Following is an explanation of this strategy: It is predicted that results from the multi-target polynomial regression model will give an integrated and functioning model. Multiple characteristics, including sensitivity, consistency, and test accuracy, will be used to create classification algorithms in line with this formula. The ambiguity matrix is what we use to gauge the indices. This matrix is a useful tool for assessing the performance of a device in terms of data recognition or measurement. The primary diagonal of the matrix should include a significant number of observations, with zero or null values in the other diagonals.

PF = Falsely Recognized positive data points

NF = Falsely Recognized Negative data points

NP = Truly Recognized negative data points

PP = Truly Recognized Positive data points

Accuracy = PP+NP / PP+NP+NF +PF (4.1)

Sensitivity = PP / PP+PF (4.2)

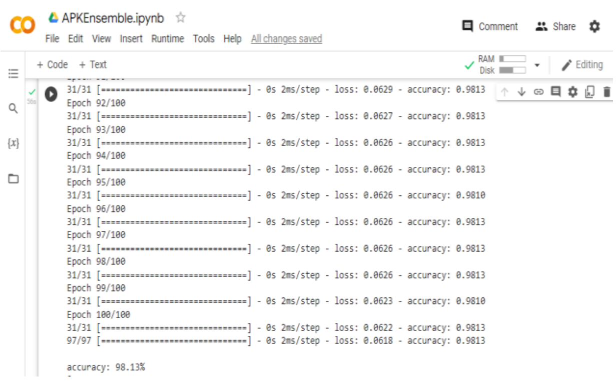
Precision = PP / PP+NF (4.3)

Specificity = NP / NP+NF (4.4)

For each class code, the table displays the values of four indices. The values were determined with the uncertainty matrix for the C5.0 algorithm. The rate of error or misclassification may also be determined on the basis of the precision index (Form 5).

Error Rate = 1 – accuracy (4.5)

Sample APKEnsemble data with their Accuracy:



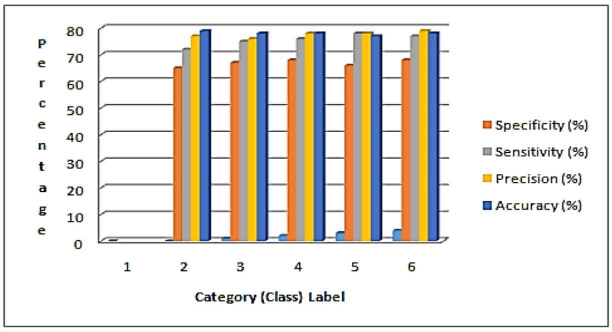


Figure 6.2 Assorted Evaluations

Table 6.1 Overall comparison of proposed system with existing system

Different algorithms	Performance in Percentage
ID3 Approach	62
Naïve Bayes Approach	91
CART Algorithm	82
Neural Network	93
J48 Approach	83
Projected algorithm	95

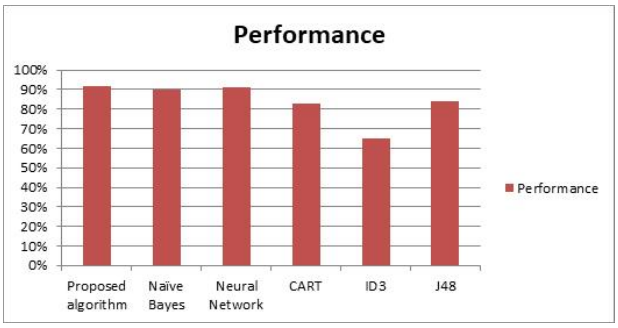


Figure 6.3 Performance Evaluations

7. Conclusion:

Mobiles and computer systems can be damaged and destroyed by malware, which is a shorthand for "malicious software." Viruses, worms, Trojan horses, spyware, adware, and ransomware are some examples of prevalent malware.Malware has expanded so quickly in recent years that the syndicates have invested extensively in technology to avoid traditional security, prompting anti-malware groups/communities to develop more strong software. It is critical to determine whether or not any particular file or piece of software has malicious code in order to keep your computer safe from a malware attack.Ensembles tend to perform better when there is a lot of variation in the models. To ensure that the models they combine are diverse, several ensemble techniques are designed in this way. More random algorithms (such as random decision trees) can yield a stronger ensemble than extremely purposeful algorithms, despite the fact that this may be counterintuitive (like entropy-reducing decision trees). A range of strong learning algorithms has been demonstrated to be more successful than strategies that aim to dumb down models in order to foster diversity. Correlation for regression tasks or information variables like cross entropy for classification tasks might boost model variety during training. When it comes to ensemble prediction accuracy, there is a dearth of research into how many classifiers are needed to make a forecast. Online ensemble classifiers need to be able to determine the size of the ensemble and the volume and velocity of the huge data streams a priori. When it came to figuring out how many parts went into a system, statistical analysis was the go-to method. It has recently been hypothesised that there is an optimal number of component classifiers for an ensemble, and having more or less of these classifiers might have a detrimental effect. "The law of decreasing returns in ensemble creation" is the name given to this phenomenon. Using the same number of independent component classifiers as class labels, their theoretical framework shows the highest accuracy.

Acknowledgments

It is not only customary but necessary for a researcher to mention his/her indebtedness to those who had helped in carrying outand enhance the research work. I pay my deep regards to God, my Parents and my loving Friends for their support and wisheswhich made this tedious work easy and successful. Finally, I would like to extend my thanks to all those who have contributed,directlyorindirectlytomakethisprojectsuccessful

References:

[1] EliteDataScience.2021.HowtoHandleImbalancedClassesinMachineLearning.[online],https://elitedatascience.com/imbalanced-classes2021

[2] Statista.2021.Mobile OS market share 2021—Statista. [online] Available at: ¡https://www.statista.com/statistics/272698/global-marketshare-held-by-mobile-operating-systems-since-2009/¿[Accessed 20 April2021].

[3] Wen, Long and Yu, Haiyang (2017) "An Android malware detection system based on machine learning" AIP ConferenceProceedings1864(1):020136.

[4] Khariwal, Kartik and Singh, Jatin and Arora, Anshul (2020) "IPDroid: Android Malware Detection using Intents andPermissions"IEEEAccess8(1):194729–194740

[5] Bai, Hongpeng and Xie, Nannan and Di, Xiaoqiang and Ye, Qing(2020) "AFAMD: A Fast Multifeature Android MalwareDetectionFramework,Design,andImplementation"IEEEAccess(8):194729–194740

[6] Mahindru, Arvind and Sangal, AL (2020) "MLDroid—framework for Android malware detection using machine

learningtechniques"NeuralComputingandApplications:1–58

[7] Wang, Ji and Jing, Qi and Gao, Jianbo and Qiu, Xuanwei (2020) "SEdroid: A robust Android malware detector usingselective ensemble learning" 2020 IEEE Wireless Communications and Networking Conference (WCNC):1–5 [8] Wang, Weiand Gao, Zhenzhen and Zhao, Meichen and Li, Yidong and Liu, Jiqiang and Zhang, Xiangliang (2018) "DroidEnsemble:DetectingAndroidmalicious applicationswithensembleofstringandstructuralstaticfeatures"IEEEAccess(6):31798–31807

[9] Sun, Lichao and Li, Zhiqiang and Yan, Qiben and Srisa-an, Witawas and Pan, Yu (2016) "SigPID: significant permissionidentificationforandroidmalware detection"201611thinternationalconference onmaliciousandunwantedsoftware(MALWARE):1–8

[10] Alswaina,FahadandElleithy,Khaled(2018),"Androidmalwarepermission-basedmulti-classclassificationusingextremelyrandomizedtrees"IEEEAccess,(6):76217–76227

[11] Chang,Wei-LingandSun,Hung-MinandWu,Wei(2016),"AnAndroidbehavior-basedmalwaredetectionmethodusingmachinelearning"2016IEEEInternationalconference onsignalprocessing,communicationsandcomputing(ICSPCC):1–4.

[12] Yuan,ZhenlongandLu,YongqiangandWang,ZhaoguoandXue,Yibo(2014),"Droid-sec:deeplearninginandroidmalwaredetection"Proceedingsofthe2014ACMconferenceonSIGCOMM:371–372.

[13] Kumaran,MonicaandLi,Wenjia(2016),"Light weightmalwaredetectionbasedonmachinelearningalgorithmsandtheandroidmanifestfile"2016IEEEMITUndergraduateResearchTechnologyConference(URTC):1–3.

[14] Adebayo, OS & Aziz, NA 2019, ‗The trend of mobile malwares and effective detection techniques ', In Multigenerational Online Behavior and Media Use: Concepts, Methodologies, Tools & Applications pp. 668-682, IGI Global.

[15] Alam, MS &Vuong, ST 2013, ‗Random forest classification for Detecting Android Malware', IEEE conference on Green Computing and Communications (Green Com) and Internet of Things (I Things/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing, Beijing, pp. 663-669, August 2013.

[16] Albasir, A, Hu, Q, Al-tekreeti, M, Naik, K, Naik, N, Kozlowski, AJ & Goel, N 2020, ‗Unsupervised Detection of Anomalous Behavior in Wireless Devices based on Auto-Encoders', NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium pp. 1-7, IEEE.

[17] Anderson, HS & Roth, P 2018, ‗EMBER: An Open Dataset for Training Static PE Malware Machine Learning Models', arXiv preprint arXiv:1804.04637.

[18] Batista, FK, Martin del Rey, A &Queiruga-Dios, A 2020, ‗A New Individual-Based Model to Simulate Malware Propagation in Wireless Sensor Networks', Mathematics, vol. 8(3), p. 410.

[19] VOC2012 accessed from, http://host.robots.ox.ac.uk/pascal /VOC/ voc2012/, accessed on April 2018.

[20] Cha, SC, Hung, SC, Chen, JF, Syu, SC & Tsai, TY 2017, ‗On the design of a blockchain-based reputation service for android applications', Advanced Science Letters, vol. 23(3), pp. 2179-2184.

[21] Chen, S, Xue, M, Fan, L, Ma, L, Liu, Y & Xu, L 2019, ‗How Can We Craft Large-Scale Android Malware? An Automated Poisoning Attack', In 2019 IEEE 1st International Workshop on Artificial Intelligence for Mobile (AI4Mobile) pp. 21-24, IEEE.

[22] Corporate News Reports, ‗Malware Reports', 2019 Published. [Online]. Available: https://www. kaspersky. com/about/press-releases. [Accessed: 8-May-2019]

[23] Emm, D, Garnaeva, M, Unuchek, R, Makrushin, D & Ivanov, A 2014, it threat evolution in Q3 2015.

[24] Dovom, EM, Azmoodeh, A, Dehghantanha, A, Newton, DE, Parizi, RM &Karimipour, H 2019, ‗Fuzzy pattern tree for edge malware detection and categorization in IoT', Journal of Systems Architecture, vol. 97, pp. 1-7.

[25] Gascon, F, Yamaguchi, D, Arp &Rieck, K 2013, Structural Detection of android Malware using Embedded call graphs‘, Conference on Artificial intelligence and security, pp. 45-54.

[26] Hsien-De Huang, T & Kao, HY 2018, R2-d2: Color-inspired convolutional neural network (CNN)-based android malware detections‘, In 2018 IEEE International Conference on Big Data (Big Data) pp. 2633-2642, IEEE.

[27] Idika& Mathur, AP 2007, A Survey of Malware Detection Techniques,‘ Department of Computer Science, Purdue University, West Lafayette, pp. 1-48.

[28] Liu, K, Xu, S, Xu, G, Zhang, M, Sun, D & Liu, H 2020, A Review of Android Malware Detection Approaches Based on Machine Learning‘, IEEE Access, vol. 8, pp. 124579-124607.

[29] Pan, Y, Ge, X, Fang, C & Fan, Y 2020, A Systematic Literature Review of Android Malware Detection Using Static Analysis. IEEE Access, vol. 8, pp. 116363-116379.

[30] Saracino, A, Sgandurra, D, Dini, G & Martinelli, F 2018, Madam: Effective and efficient behavior-based android malware detection and prevention. IEEE Transactions on Dependable and Secure Computing, vol. 15(1), pp. 83-97.

[31] Su, D, Liu, J, Wang, X & Wang, W 2019, Detecting Android Locker-Ransomware on Chinese Social Networks. IEEE Access, 7, 20381- 20393.

[32] Statista Official, 'Statistical Analysis and Reports', 2019 Published. [Online]. Available: https://www. statista. com/statistics/271774/shareof-android-platforms-on-mobile-devices-with-android-os/. [Accessed: 10-May-2019].

[33] Victor, ‗Mobile Malware Evolution', 2019 Published. [Online]. Available: https://securelist. com/mobile-malware-evolution-2018 /89689 [Accessed: 8-May-2019].

[34] Zhou, Y & Jiang, X 2012, Dissecting android malware: Characterization and evolution. In 2012 IEEE symposium on security and privacy (pp. 95-109, IEEE).

[35] Zhu, HJ, You, ZH, Zhu, ZX, Shi, W. L, Chen, X & Cheng, L 2018, DroidDet: effective and robust detection of android malware using static analysis along with rotation forest model. Neurocomputing, vol. 272, pp. 638-646.

[36] Yang, XS 2010, Nature-inspired metaheuristic algorithms. Luniver press, 2010.

[37] Wang, W, Zhao, M & Wang, J 2019, Effective android malware detection with a hybrid model based on deep autoencoder and convolutional neural network. Journal of Ambient Intelligence and Humanized Computing, 10(8), 3035-3043.

[38] Tong, F & Yan, Z 2017, ‗A hybrid approach of mobile malware detection in Android‘, Journal of Parallel and Distributed computing, vol. 103, pp. 22-31.

[39] Kothari,SonaliandKarde,PravinandThakare, Vilas(2018)"StaticAnalysisofAndroidPermissionsandSMSusingMachineLearningAlgorithms"InternationalJournalofComputerApplications975:888

[40] Nisa, M.; Shah, J.H.; Kanwal, S.; Raza, M.; Khan, M.A.; Damaševiˇcius, R.; Blažauskas, T. Hybrid malware classification method using segmentation-based fractal texture analysis and deep convolution neural network features. Appl. Sci. 2020, 10, 4966. [CrossRef]

[41] Amin, M.; Shah, B.; Sharif, A.; Ali, T.; Kim, K.l.; Anwar, S. Android malware detection through generative adversarial networks. Trans. Emerg. Telecommun. Technol. 2019, e3675. [CrossRef]

[42] Arp, D.; Spreitzenbarth, M.; Hubner, M.; Gascon, H.; Rieck, K.; Siemens, C. Drebin: Effective and explainable detection of android malware in your pocket. In Proceedings of the 2014 Network and Distributed System Security Symposium, San Diego, CA, USA, 23–26 February 2014; doi:10.14722/ndss.2014.23247 [CrossRef]

[43] Google Play. Available online: https://play.google.com/ (accessed on 19 May 2021).

[44] AndroZoo. Available online: https://androzoo.uni.lu/ (accessed on 19 May 2021).

[45] AppChina. Available online: https://tracxn.com/d/companies/appchina.com (accessed on 19 May 2021).

[46] Tencent. Available online: https://www.pcmgr-global.com/ (accessed on 19 May 2021).

[47] YingYongBao. Available online: https://android.myapp.com/ (accessed on 19 May 2021).

[48] Contagio. Available online: https://www.impactcybertrust.org/dataset_view?idDataset=1273/ (accessed on 19 May 2021).

[49] Zhou, Y.; Jiang, X. Dissecting android malware: Characterization and evolution. In Proceedings of the 2012 IEEE Symposium on Security and Privacy, San Francisco, CA, USA, 20–23 May 2012; pp. 95–109. [CrossRef]

[50] VirusShare. Available online: https://virusshare.com/ (accessed on 19 May 2021).

[51] Intel Security/MacAfee. Available online: https://steppa.ca/portfolio-view/malware-threat-intel-datasets/ (accessed on 19 May 2021).

[52] Chen, K.; Wang, P.; Lee, Y.; Wang, X.; Zhang, N.; Huang, H.; Zou, W.; Liu, P. Finding unknown malice in 10 s: Mass vetting for new threats at the google-play scale. In Proceedings of the 24th USENIXSecurity Symposium (USENIX Security 15), Redmond, WA, USA, 7–8 May 2015; pp. 659–674.

[53] Android Malware Dataset. Available online: http://amd.arguslab.org/ (accessed on 19 May 2021).

[54] APKPure. Available online: https://m.apkpure.com/ (accessed on 19 May 2021).

[55] Anrdoid Permission Dataset. Available online: https://data.mendeley.com/datasets/b4mxg7ydb7/3 (accessed on 19 May 2021).

[56] Maggi, F.; Valdi, A.; Zanero, S. Andrototal: A flexible, scalable toolbox and service for testing mobile malware detectors. In Proceedings of the Third ACM Workshop on Security and Privacy in Smartphones & Mobile Devices, Berlin, Germany, 8 November 2013; pp. 49–54. [CrossRef]

[57] Wandoujia App Market. Available online: https://www.wandoujia.com/apps (accessed on 19 May 2021).

[58] Google PlaystoreAppsin Kaggle. Available online: https://www.kaggle.com/gauthamp10/google-playstore-apps (accessed on 19 May 2021).

[59] CICMaldroid Dataset. Available online: https://www.unb.ca/cic/datasets/maldroid-2020.html (accessed on 19 May 2021).

[60] AZ Dataset. Available online: https://www.azsecure-data.org/other-data.html/ (accessed on 19 May 2021). Electronics 2021, 10, 1606 32 of 34.

[61] Wang, W, Zhao, M & Wang, J 2019, Effective android malware detection with a hybrid model based on deep autoencoder and convolutional neural network. Journal of Ambient Intelligence and Humanized Computing, 10(8), 3035-3043.

[62] Github Malware Dataset. Available online: https://github.com/topics/malware-dataset (accessed on 19 May 2021).

[63] Wang, S, Chen, Z, Yan, Q, Yang, B, Peng, L & Jia, Z 2019, A mobile malware detection method using behavior features in network traffic. Journal of Network and Computer Applications, 133, 15-25.

[64] Wang, W, Li, Y, Wang, X, Liu, J & Zhang, X 2018, Detecting Android malicious apps and categorizing benign apps with ensemble of classifiers. Future Generation Computer Systems, 78, 987-994.

[65] Victor, 'Mobile Malware Evolution', 2019 Published. [Online]. Available: https://securelist. com/mobile-malware-evolution-2018 /89689 [Accessed: 8-May-2019]

[66] Tangil, S, Tapiador, JE, Lopez, PP &Ribagorda, A 2014, 'Evolution, Detection and Analysis of Malware for Smart Devices', IEEE Journal of Communications Surveys & Tutorials, vol. 16, issue 2, pp.961-987.

[67] Tong, F & Yan, Z 2017, 'A hybrid approach of mobile malware detection in Android', Journal of Parallel and Distributed

computing, vol. 103, pp. 22-31.

[68] Valli & Brand, M 2008, 'The Malware Analysis Body of Knowledge (MABOK)', Proceedings of the 6th Australian Digital Forensics Conference, Australia, pp. 1-7.

[69] Suarez-Tangil, G, Dash, SK, Ahmadi, M, Kinder, J, Giacinto, G & Cavallaro, L 2017, DroidSieve: Fast and accurate classification of obfuscated android malware. In Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy (pp. 309-320, ACM.

[70] Tam, K, Feizollah, A, Anuar, N. B, Salleh, R & Cavallaro, L 2017, The evolution of android malware and android analysis techniques. ACM Computing Surveys (CSUR), 49(4), 76.

[71] Tamersoy, K. Roundy & Chau, DH 2014, 'Guilt by association: large scale malware detection by mining file-relation graphs', ACM 20th International Conference on Knowledge Discovery and Data Mining, pp. 1524-1533.

[72] Onwuzurike, L, Mariconti, E &riotis, P, Cristofaro, E. D, Ross, G &Stringhini, G 2019, 'MaMaDroid: Detecting android malware by building markov chains of behavioral models (extended version), ACM Transactions on Privacy and Security (TOPS), vol. 22(2), p. 14.

[73] Ouaguid, A, Abghour, N &Ouzzif, M 2018, 'A novel security framework for managing Android permissions using blockchain technology', International Journal of Cloud Applications and Computing (IJCAC), vol. 08(1), pp. 55-79.

[74] Overton, M 2005, 'Anti-Malware Tools: Intrusion Detection Systems', EICAR conference, Malta, pp. 1-22.

[75] Pan, Y, Ge, X, Fang, C & Fan, Y 2020, A Systematic Literature Review of Android Malware Detection Using Static Analysis. IEEE Access, vol. 8, pp. 116363-116379.

AuthorsProfile

SumalthaPotteti is working as Assistant Professor at Bhoj Reddy Engineering College for Women, Hyderabad,Telangana (State), India. She has received B.Tech (CSE), M.Tech (CSE) Degree in Computer Science and Engineering. Her main researchinterestincludesMachineLearningandIntrusionDetection.

Dr. G.S. Mahalakshmi is working as Associate Professor at Anna University, Guindy, Chennai,Tamilnadu (State), India. She has received B.E(CSE), M.E(CSE), Ph.D. Degree in Computer Science and Engineering.Her main research interest includes Artificial Intelligence, Machine Learning, Natural Language Processing knowledge representation and Text Mining.