Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 3, Issue 5, 2025

ANDROID MALWARE ANALYSIS USING ARTIFICIAL INTELLIGENCE

Ali Ahmed[1], Noman Khokhar*[2], Nelson Alfonso[3], Karan Kumar[4]

[1]MSCS, Department of Computer Science, Sindh Madressatul Islam University, Karachi, Sindh, Pakistan
*[*2]MSCS, Department of Computer Science, Iqra University, Karachi, Sindh, Pakistan*
[3] Masters of Science in computing (Software Engineering), the Open University, United Kingdom
[4]Masters in Quality and Production Management, Częstochowa University of Technology, Poland

[1]aliahmedreal@gmail.com, [*2]nomankhokhar29@gmail.com, [3]nelson_alfonso@icloud.com, [4]karan.chabria1@gmail.com

DOI: https://doi.org/10.5281/zenodo.15331551

Keywords
Android, Malware, Neural Networks, Machine Learning, Static Analysis, Dynamic Analysis, KNN, Decision Tree, NB, Deep Neural Nets.

Article History
Received on 26 March 2025
Accepted on 26 April 2025
Published on 03 May 2025

Corresponding Author: *
Noman Khokhar*[2]

Abstract

Mobile phones have become a crucial part of society and serve as more than just communication devices. The growing use of smartphones has led to a large number of apps, making it difficult for app marketplaces to validate their legitimacy. Conventional security solutions for computer malware are challenging to apply on mobile devices due to different resource management mechanisms. Implementing intelligent tools using the Machine Learning in the threat identification process of security software can improve its efficiency by analyzing data and identifying potential threats. This reduces the need for human intervention and allows for faster detection of risks, saving time and resources. Intelligent tools can also continuously monitor data and identify potential threats in real-time, further improving the threat identification process. In conclusion, the use of intelligent tools can significantly enhance the effectiveness of conventional security software and protect against potential threats. This can help prevent hacking and data theft and keep personal information safe and secure. Additionally, these intelligent tools can be easily integrated into current security systems, making it easy for organizations to improve their overall security posture.

INTRODUCTION

After the initial conception by Cohen in 1983, the term "computer virus" has undergone development and now encompasses the detection of various forms of malware. Mobile phones have become integral to modern society, serving as more than just a communication tool but also a source of information and connectivity [1]. The widespread use of smartphones has led to a proliferation of apps, making it difficult for app marketplaces to validate the legitimacy of all apps. However, mobile devices have different resource management mechanisms compared to PCs, which can make it challenging to use traditional security solutions for computer malware on mobile devices. One way to improve the efficiency of the threat identification process in conventional security software is through the use of intelligent tools. These tools can analyze data and identify patterns that may indicate a potential threat, reducing the need for human intervention and allowing for faster identification of potential risks [2]. By implementing intelligent tools in the threat identification process, organizations can more efficiently identify and respond to potential threats, which can save time and resources. Additionally, intelligent tools can be configured to continuously monitor data and identify potential threats in real-

time, further improving the efficiency of the threat identification process. Overall, the incorporation of intelligent tools can significantly improve the effectiveness of conventional security software and enhance an organization's ability to protect against potential threats.

Android security is of utmost importance for the protection of personal and sensitive information stored on mobile de vices. Android is one of the most widely used mobile operating systems and its open-source nature makes it susceptible to various security threats such as malware, viruses, and hacking attempts. The security of Android devices plays a critical role in ensuring the confidentiality and privacy of the users' data. The importance of Android security stems from the fact that mobile devices store personal and sensitive information such as login credentials, personal information, financial data, and other sensitive files. If the device is not secured adequately, this information can be easily compromised and lead to identity theft, financial fraud, and other types of cybercrimes. Furthermore, Android devices are connected to the internet, making them vulnerable to various online threats such as phishing attacks, malware, and spyware.

The consequences of poor Android security can be severe and can have far-reaching impacts on the user's privacy and personal information. Here are some of the consequences of inadequate Android security:

A. DATA BREACHES:

One of the most significant consequences of poor Android security is data breaches. If a user's Android device is com promised, sensitive information such as login credentials, financial data, and personal information can be accessed by unauthorized individuals, resulting in identity theft, financial fraud, and other cybercrimes.

B. MALWARE AND VIRUSES:

Android devices that are not adequately secured are susceptible to malware and viruses that can corrupt the device's data and software. Malware can also compromise the user's privacy by accessing and stealing sensitive information, such as personal messages, call logs, and browsing history.

C. PHISHING ATTACKS:

Android devices that are not secured can also fall victim to phishing attacks. Phishing attacks are attempts by cybercriminals to obtain sensitive information from users by posing as a legitimate entity. These attacks can lead to the compromise of user accounts and loss of sensitive data.

D. DEVICE HIJACKING:

Poor Android security can result in device hijacking, where hackers can take control of the device, alter its settings, and steal sensitive information. This can result in the loss of privacy and control over the device, leading to potential misuse of personal information.

II. LITERATURE REVIEW

With the proliferation of smartphones, individuals have access to a wide range of information and services at their fingertips, including access to the internet, real-time news and updates, and various applications and features.

A smartphone is a type of mobile device that combines the functionality of a personal computer with the ability to make telephone calls. It typically includes a touchscreen display, a processor, memory, and storage, as well as various sensors and features such as a camera, GPS, and biometric authentication. Smartphones also have the ability to connect to the internet and run a variety of applications, or "apps," that enable users to perform a wide range of tasks such as messaging, browsing, gaming, and more. Many smartphones also have the ability to connect to other devices and accessories, such as smartwatches and headphones, to enhance their functionality and capabilities. Overall, smartphones are highly advanced and versatile devices that have become an integral part of modern society [3][4].

In the developed society, mobile phones have become a very important element. It's not just a tool, it's a "smart" part of our lives. These tools often help people answer questions like times, weather, news, sports scores, ads, and more. This gives you instant and unlimited access to information as per your need. This means that mobile phones have become a very important part of people's lives. Today, smartphones are equipped with a range of features and capabilities beyond those of their early counterparts. In addition

to basic communication functions such as calls, text messages, and emails, modern smartphones allow users to access the internet, use a variety of applications, and perform many of the same tasks as a personal computer. This includes activities such as browsing the web, creating and editing documents, managing finances, and staying connected with others through social media. As a result, smartphones have become a staple in modern society, with many individuals relying on them for both personal and professional purposes. Additionally, the cost of smartphones has decreased significantly over the years, making them more accessible to the general public [5].

The advancement of smartphone technology has also led to the integration of a variety of sensors and features, such as touchscreens, GPS, cameras, and biometric authentication. These capabilities allow for an even wider range of functions, including navigation, photography, and secure access to personal information. Moreover, the constant evolution of mobile technology has led to the development of various operating systems and platforms, such as iOS and Android, which offer users a range of customization options and app stores for downloading additional features and tools. Overall, the capabilities of modern smartphones have significantly expanded from the limited functions of early models, making them essential devices for many people in their daily lives [7].

A. OPERATING SYSTEM:

An operating system (OS) is a software platform that man ages the hardware and software resources of a device and provides an interface for users to interact with the device [6]. It serves as the foundation for running and executing various types of applications and programs on the device. An operating system typically includes a kernel, which is the central part of the system responsible for managing the hardware and software resources, as well as various libraries, utilities, and other components that enable the operation and functionality of the device [29]. It also includes a user interface, which allows users to interact with the device and access its features and functions [8]. There are various types of operating systems, such as those designed for desktop computers, servers, mobile devices, and more, each with its own unique features and capabilities [14]. An operating system is a software platform that enables the user to run various types of applications on devices such as mobile phones, tablets, and smartwatches. It provides an interface for managing the interaction between the user and the hardware of the device. There are several different types of operating systems, each with its own unique features and capabilities.

B. TYPES OF OPERATING SYSTEM:

Android: An open-source operating system developed by Google and used on a variety of devices, including smart phones, tablets, and wearables. It offers users customization options and access to a large selection of apps through the Google Play store [9].

IOS: A proprietary operating system developed by Apple and used exclusively on the company's iPhone, iPad, and iPod touch devices. It has a user-friendly interface and a variety of built-in features and apps, as well as access to the App Store for downloading additional apps [19].

BlackBerry OS: A proprietary operating system developed by BlackBerry Limited and used on BlackBerry smart phones. It is popular among business users for its ability to synchronize with various business applications and the BlackBerry Enterprise Server.

Windows Phone: A proprietary operating system developed by Microsoft and used on a range of devices, including smartphones and tablets. It offers a range of features and apps, as well as integration with other Microsoft products and services.

Symbian: A mobile operating system developed by the Symbian Foundation and used on devices such as smart phones and feature phones. It has since been discontinued in favor of other operating systems [3].

Bada: A mobile operating system developed by Samsung and used on a range of devices.

Firefox OS: An open-source operating system developed by the Mozilla Foundation and used on a range of devices.

Palm OS: A mobile operating system developed by Palm, Inc. and used on devices such as smartphones and personal digital assistants.

C. WHY CHOOSE ANDROID:

Android, the most widely-used mobile operating system in the world, has faced increased threats from malware due to its technical capabilities, open-source code, and the ability to install third party applications without central control. Despite the presence of security mechanisms, there have been numerous reports of vulnerabilities and malware attacks on Android devices [10]. To address this issue, researchers and developers have implemented various security solutions utilizing static analysis, dynamic analysis, and artificial intelligence to mitigate malware attacks [23]. The use of data science in cybersecurity has also emerged as a promising approach, as data-driven analytical models can provide insights that help to anticipate and prevent malicious activity. It is crucial to continue developing and improving these methods and systems to enhance the security of Android devices [13].

Wide Adoption: Android has a very large market share, with a significant portion of the global smartphone market using Android devices. This has led to a large and active developer community, which has contributed to the development and improvement of the operating system [16].

Customization options: Android is an open-source operating system, which means that it can be modified and customized by developers and device manufacturers [12]. This allows for a wide range of customization options, including the ability to add new features and functionality [14]. Large app ecosystem: Android has a large and diverse selection of apps available through the Google Play store, which offers a wide range of functionality and caters to awide range of users [17].

Compatibility with other devices: Android devices are compatible with a wide range of other devices and accessories, such as smartwatches and headphones, which can enhance the functionality of the device [18]- [21].

Regular updates: Android releases regular updates to improve the performance and security of the operating system,as well as to add new features and functionality. Overall, the combination of these factors has contributed to the success and popularity of Android as a powerful mobile operating system [11].

To further protect against malware attacks on Android devices, it is important to regularly update the operating system and install security patches as they become available. Using a reputable antivirus software and only downloading apps from trusted sources can also help reduce the risk of infection. Additionally, implementing strong passwords and enabling two-factor authentication can provide an extra layer of security for device access and online accounts [30]. It is essential for individuals and organizations to be proactive in maintaining the security of their Android devices, as the consequences of a successful malware attack can be severe, including loss of sensitive data, financial damage, and disruption of operations [28]. Therefore, it is advisable to follow best practices and adopt a holistic approach to cybersecurity to ensure the protection of Android devices [15].

III. COMPARISON OF ANDROID WITH OTHER OPERATING SYSTEMS:

In this section we are comparing the Android OS with different operating systems, below mentioned are the detail:

A. ANDROID VS. IOS:

Android is an open-source operating system, while iOS is proprietary and only used on Apple devices. This means that Android can be modified and customized by developers and device manufacturers, while iOS cannot [19]. Android has a larger and more diverse app ecosystem, with a wider range of apps available through the Google Play store. iOS has a smaller selection of apps available through the App Store, but these apps are often of higher quality and go through a more stringent review process. Android devices tend to have a lower price point compared to iOS devices, which may make them more attractive to budget-conscious consumers.

Android and iOS have similar user interfaces and offer similar features and functionality, but there are some differences in design and user experience [20].

B. ANDROID VS. BLACKBERRY OS:

Android is an open-source operating system, while Black Berry OS is proprietary and only used on BlackBerry devices. Android has a larger and more diverse app ecosystem, with a wider range of apps available through the Google Play store [15]. BlackBerry OS has a smaller selection of apps available through the BlackBerry World app store. Android devices tend to have more advanced hardware and features compared to BlackBerry devices, which may make them more appealing to users looking for a high-performance device. BlackBerry OS is popular among business users due to its ability to synchronize with various business applications and the BlackBerry Enterprise Server, while Android has a broader consumer base [22].

C. ANDROID VS. WINDOWS PHONE:

Android is an open-source operating system, while Windows Phone is proprietary and used on a range of devices including smartphones and tablets. Android has a larger and more diverse app ecosystem, with a wider range of apps available through the Google Play store [23] Windows Phone has a smaller selection of apps available through the Microsoft Store. Android devices tend to have more advanced hardware and features compared to Windows Phone devices, which may make them more appealing to users looking for a high-performance device [29]. Windows Phone integrates seamlessly with other Microsoft products and services, which may make it more appealing to users who are heavily invested in the Microsoft ecosystem.

IV. ANDROID SYSTEM ARCHITECTURE:

The Android operating system, which is based on the Linux kernel, utilizes a software stack to establish its hierarchical system architecture. Google provides a classic layered architecture of the Android system, as depicted in the figure, that is arranged from bottom to top and includes the Linux kernel, hardware abstraction layer, native C++/C libraries, Android Runtime environment, Java API framework, and the application layer. Each layer encompasses a multitude of submodules and subsystems that work together to form the complete system. The kernel space located at the bottom of the Android stack is the foundation and cornerstone of the entire system, and it is comprised of the Linux kernel [24]. On the other hand, the user space found at the top of the Android system is made up of native C++/C libraries, the Android Runtime, and the Java API framework, which are all important components that work in harmony to deliver a seamless user experience. The communication between the kernel and user spaces is facilitated by system calls [28].

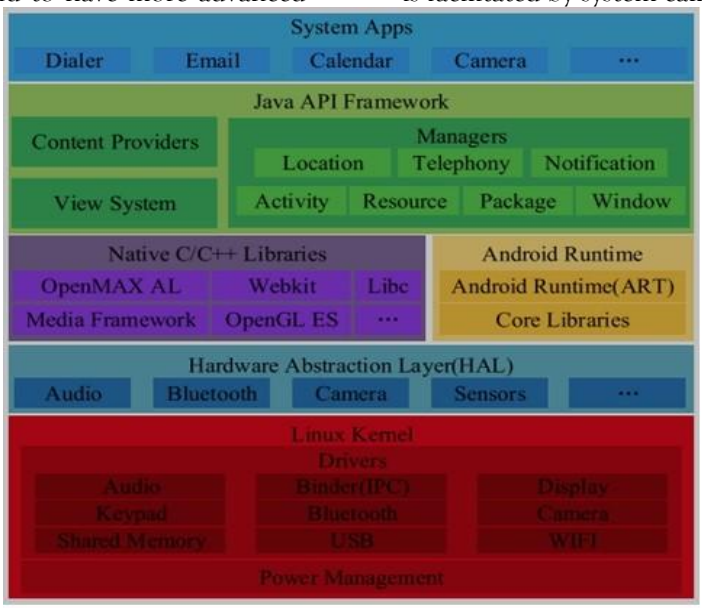


Figure explain the Android device log management, the portion in the red block highlighted the complex end user interfaces. User space programs are predominantly written in either C++ or Java, and these two programming languages play a crucial role in the functioning of the Android system. The Java native interface acts as a bridge between the Java layer and the native layer of the user space, connecting them to the rest of the Android system. In conclusion, the Android system architecture is well-structured, with each layer and component working together to deliver a robust and seamless experience to the end-user [16].

V. ANDROID SECURITY MECHANISMS:

The Android operating system functions as a divided privileges system, utilizing Binder, an inter process communication mechanism, to implement a suite of system services and distinguish applications through unique Linux UIDs. By de fault, applications on Android are given limited permissions, but they can request additional ones for interaction with the system services, hardware, and other applications [15]. The permissions that an Android app requires are specified in the Androidmanifest.xml file and are granted either during installation or at runtime. The system uses UIDs to manage and enforce the permissions granted to each app [17].

The security of the Android operating system is considered of utmost importance by its developers, who have made significant efforts to enhance it through updates and advancements. One example of this is the release of Android Q in 2019, which introduced several new security features such as file-based encryption, improved access control for sensitive 4 VOLUME1,2023 information, advanced permission control, and more [17]. Despite these efforts, the Android ecosystem still faces the threat of malware. To address this issue, the developers continue to work on improving the security features of the Android operating system [25].

VI. RESEARCH METHODOLOGY:

The proposed model employs a classifier with hyperparameters optimized using grid search, which allows for effective classification of the packets into legitimate and malicious categories. The captured preprocessed logs are transformed into a structured data format, such as a data frame, to facilitate efficient data analysis [26]. The data is then processed to generate a heatmap based on cosine similarity, which is a measure of the similarity between two vectors. In this case, the vectors represent packet logs, and the cosine similarity is used to identify patterns of similarity between legitimate and malicious packets [17]. The proposed model uses feature classifiers to differentiate between legitimate and malicious packets. Feature classifiers are algorithms that are used to identify specific features in data that are associated with a particular class. In this case, the features are specific patterns in the packet logs that are indicative of either legitimate or malicious activity. To train the model, a set of labeled data is required, where each packet is labeled as either legitimate or malicious. The model uses the labeled data to learn the patterns that are as sociated with each class of packets. The model is then tested on a separate set of labeled data to evaluate its performance [26]

VII. DATA SETS:

To ensure a comprehensive analysis, I conducted extensive research across multiple platforms to gather a diverse range of relevant data sets that would be used to train and test my model. The following datasets were utilized during the model training phase to ensure a robust and accurate model.

A. DREBIN:

To support ongoing research on Android malware and facilitate the comparison of different detection methods, the Drebin project has made its dataset publicly accessible. This dataset includes a total of 5,560 applications from 179 unique malware families and has been compiled over the time frame from August 2010 to October 2012. These samples were then subjected to systematic analysis to understand their various aspects, including the methods of installation, the mechanisms of activation, and the nature of the malicious payloads they carried [11]. The analysis and subsequent evolutionary study of representative families showed that these malware strains are evolving rapidly in an attempt to evade detection by current mobile antivirus software. Testing with four different mobile security software in November 2011 demonstrated

that the best performing software was able to detect 79.6% of the samples, while the lowest performing software detected only 20.2% of the samples [16]. These results indicate a need for the development of next-generation anti-malware solutions that can more effectively protect against Android malware. The samples for this dataset were provided by the Mobile Sandbox project. The availability of this dataset allows researchers to more easily study and com pare various approaches to detecting and mitigating Android malware threats.

In addition to the raw data, technical details about the dataset can be found in the corresponding research paper. By making this information readily available to the research community, it is hoped that the development of more effective methods for detecting and defending against Android malware can be accelerated [29].

B. VIRUS SHARE:

VirusShare is a repository of malware samples that is designed to provide access to live malicious code for security researchers, incident responders, forensic analysts, and those with a general interest in this type of information. The repository is intended to serve as a resource for those who need access to samples of live malware for research, analysis, or other purposes. By making these samples available, VirusShare aims to support the efforts of those working to understand and defend against malicious code and to help improve the overall security posture of organizations and individuals. In addition to serving as a source of samples, VirusShare also provides various tools and resources to assist researchers in their work, such as documentation, analysis tools, and other helpful materials. Overall, VirusShare is an important resource for those working in the field of cybersecurity and a valuable resource for those looking to understand and defend against malware threats [26].

TABLE 1. Training Data

DataSet	Total	Benign	Malicious
VirusShare	200000	100000	100000
Derbin	450000	225000	225000
Kaggle	900000	300000	600000

TABLE 2. Testing Data

DataSet	Total	Benign	Malicious
VirusShare	40000	21000	19000
Derbin	195000	97500	97500
Kaggle	280000	140000	140000

TABLE 3. Validation Data

DataSet	Total	Benign	Malicious
VirusShare	14000	5000	9000
Derbin	20000	10000	10000
Kaggle	170000	90000	80000

VIII. APPROACH OF ANALYSIS:

There are various approaches that can be used to analyze and mitigate cyber threats on Android devices. Some of the common approaches include: Static analysis: This involves analyzing the code of an app or system without executing it, in order to identify vulnerabilities or malicious elements.

Dynamic analysis: This involves analyzing the behavior of an app or system while it is running, in order to detect any malicious activity or anomalies.

Machine learning: This involves using algorithms and statistical models to analyze and classify data, in order to identify patterns and trends that can help predict and prevent cyber threats. **Sandboxing:** This involves creating a separate, isolated environment in

which an app or system can be tested and analyzed, in order to prevent any potential harm to the main system [28].

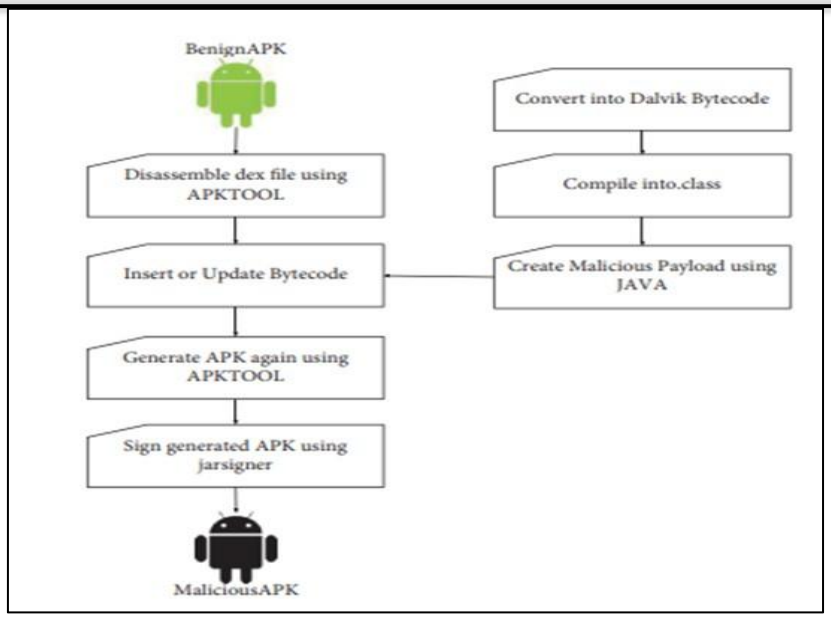
Reverse engineering: This involves deconstructing and analyzing an app or system in order to understand its inner workings and identify vulnerabilities or malicious elements. By using a combination of these approaches, organizations and security professionals can effectively analyze and mitigate cyber threats on Android devices.

We will need to carefully analyze a set of malware and benign samples to identify any differences in the permissions used in order to further explore this hypothesis. This analysis will involve closely examining the characteristics of each sample, including the specific permissions that are requested and granted. By thoroughly examining these features, we will be able to determine whether there is a significant differential in the permissions used between the malware and benign samples. In addition to analyzing the permissions used, we may also want to consider other factors that could potentially impact the results of this study. This could include the specific operating system or device on which the samples are being analyzed, as well as any other software or security measures that may be in place. By taking these additional factors into account, we will be able to provide a more comprehensive and accurate analysis of the hypothesis. Overall, this first step is critical in determining whether there is a differential in the permissions used between malware and benign samples. By thoroughly examining these features and considering any potential influencing factors, we will be able to provide a more complete and accurate analysis of this important hypothesis [23]. The next analysis will involve the examination of the Malgenome dataset. As previously mentioned, there are now numerous sources of examples and malware families that can be useful for future research. The objective of the next experiment and analysis is to present our initial findings [27]. In order to broaden the scope of the study, an additional 90% of the Malgenome dataset will be included in the analysis. This will provide a more comprehensive view of the current state of Android malware and facilitate the production of more detailed and accurate results [21]. The 'Type' label indicates whether an application is mal ware or not, and as can be seen, this dataset is balanced. To gain insights into the characteristics of the malware samples, we will now analyze the top 10 permissions used in these samples.

The previous outputs provide us with valuable insights into the distinctions in permissions utilized between malware and benign applications. By examining the data, we are able to gain a better understanding of the ways in which these types of software differ in terms of their permissions, which can be useful in identifying and detecting malware in the future [24].

IX. MODELING:

The recent results demonstrate how various classifiers can be trained to detect malware using its permissions. However, as mentioned earlier, this is only a preliminary analysis, and further optimization of the classifiers could be achieved through the adjustment of hyperparameters and other variables. While these initial results are promising, it is important to note that this investigation is only a first step, and further work is needed to more fully understand how these classifiers perform and how their accuracy can be improved. By exploring a wider range of parameters and options, we can better understand the capabilities and limitations of these classifiers and work towards more effective malware detection methods. In order to thoroughly evaluate the performance of these classifiers, it will be necessary to conduct additional experiments and analyses, utilizing a range of datasets and parameters to fully understand their capabilities and limitations [22].



X. DYNAMIC ANALYSIS:

For this study, we utilized a set of pcap files from the DroidCollector project comprising 4705 benign and 7846 malicious applications. All of the files were processed using our feature extraction script. The purpose of this analysis 6 VOLUME1, 2023 is to determine if it is possible to distinguish between mal ware and benign applications based on their network traffic patterns, as previous static analysis has shown that many applications utilize network connections for communication or data transmission. By analyzing the traffic patterns of these applications, we aim to answer the question of whether it is possible to differentiate between malicious and benign software using network traffic data. In order to more fully understand the capabilities and limitations of this approach, it will be necessary to conduct additional experiments and analyses, utilizing a range of datasets and parameters to fully evaluate the performance of this method. In particular, further investigation is needed to determine how the size and complexity of the dataset affects the accuracy of the results, as well as how the various features and parameters of the network traffic data can be utilized to improve the performance of the classifiers [21].

XI. RESULTS AND FIGURES:

Dependency of different packets over the network. When an Android device connects to a network, a series of technical steps occur behind the scenes. The device first scans for available Wi-Fi networks in the area by sending out a series of probe requests on each Wi-Fi channel to discover any nearby access points (APs) [10].

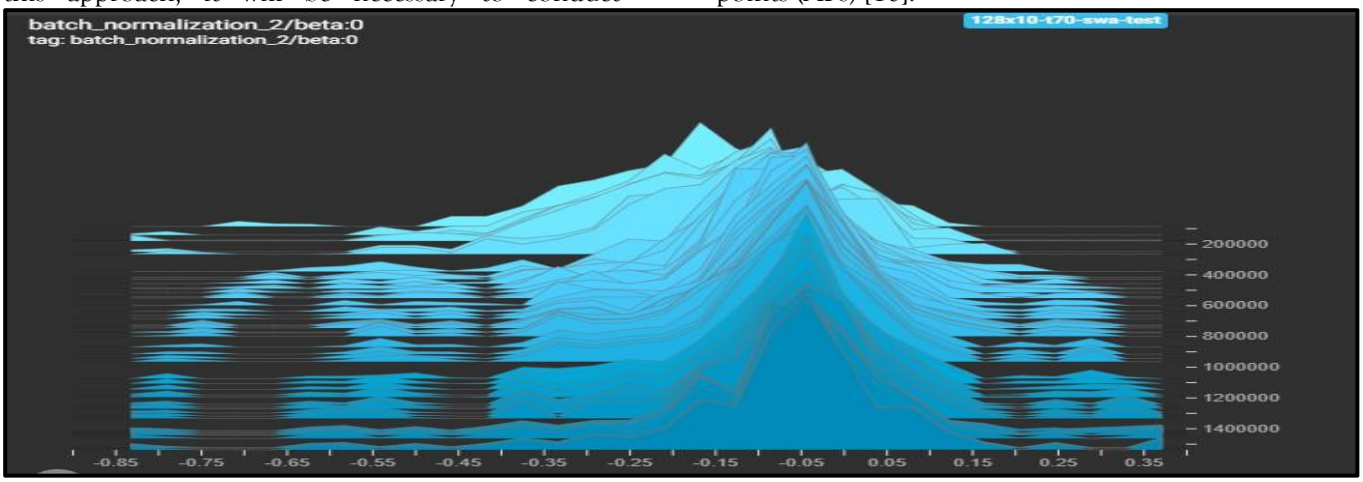
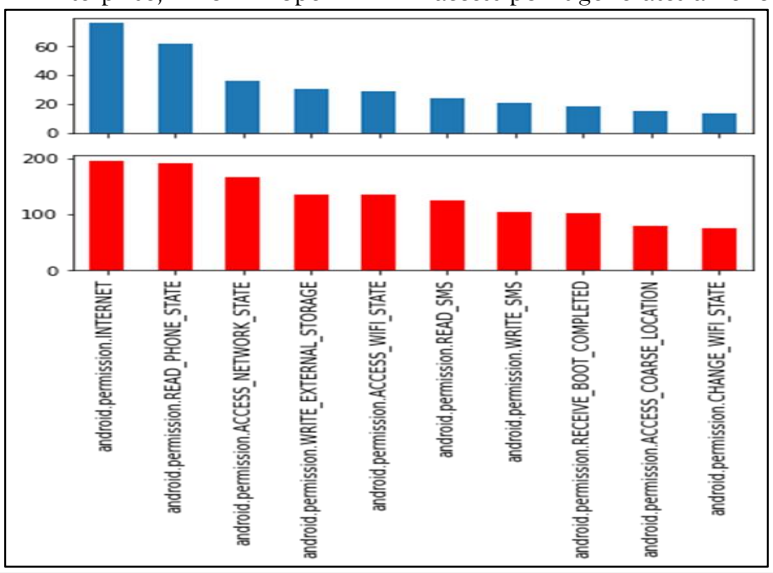


FIGURE 1. Normally distributed Histogram

Once the device finds an available network, it sends an authentication request to the access point using a variety of authentication methods, including WPA2-PSK, WPA2- Enterprise, or open authentication. If the access point re- quires a password or other credentials, the device prompts the user to enter them. Once authenticated, the access point generates an encryption key [5].



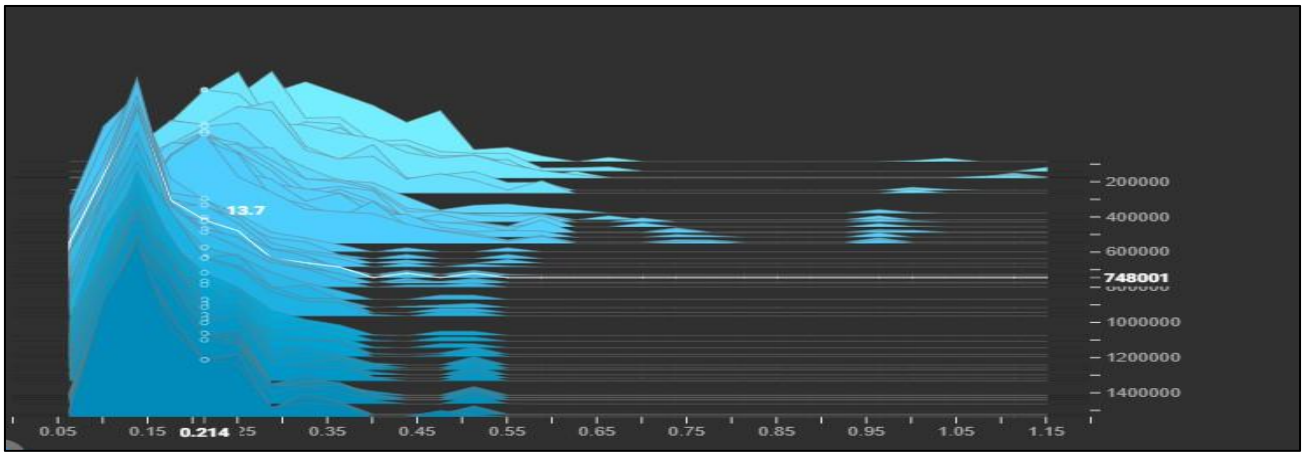
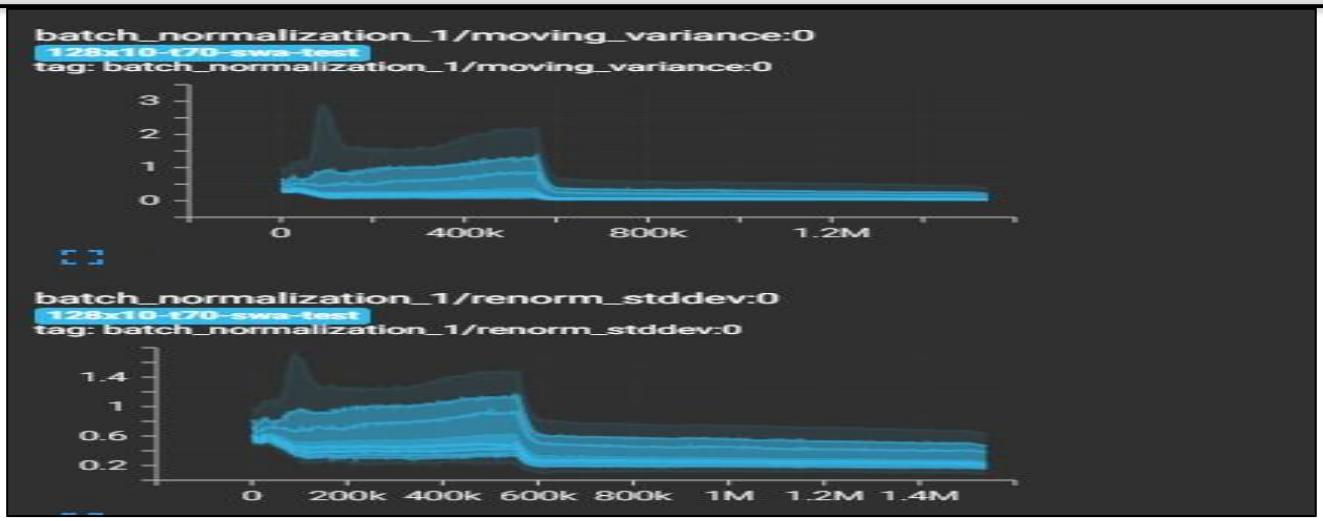


FIGURE 2. Data Visualization: As per the classification.

In the EDA below mentioned are the plots of batch normalization:

- **Moving variance()**
- **Renorm stddev()**



Histograms log of tensor board, showing the model fitting over the newly distributed data. The proposed model uses feature classifiers to differentiate between legitimate and malicious packets. Feature classifiers are algorithms that are used to identify specific features in data that are associated with a particular class. In this case, the features are specific patterns in the packet logs that are indicative of either legitimate or malicious activity [20].

In order to implement the model, a significant amount of computational power is necessary. To achieve this, one can make use of graphical processing units and cloud instances. The forward approach for malware detection that relies on code semantic features. The authors employed a graph con- volutional network (GCN) to extract these features, which provide advanced semantic information for the classification of applications. The accuracy of the proposed approach can be further improved by integrating additional features with the semantic features [16].

TABLE 4. Accuracy of different models

Name	DNN	ANN	SVM	NB	DT	LR	KNN
Derbin	91.8	70.5	70.5	74.8	74.8	35.6	29.3
VirusShare	87.9	68.6	68.8	71.4	73.4	28.4	39.8
Kaggle	92.4	72.2	83.9	86.5	74.2	54.1	41.4

The proposed model uses feature classifiers to differentiate between legitimate and malicious packets. Feature classifiers are algorithms that are used to identify specific features in data that are associated with a particular class. In this case, the features are specific patterns in the packet logs that are indicative of either legitimate or malicious activity [1].

As shown in the figure validation accuracy increases along with the training accuracy, it suggests that the model is generalizing well and is not overfitting to the training data. The training accuracy indicates how well the model is fitting the training data, while the validation accuracy indicates how well the model is generalizing to new, unseen data. If both training and validation accuracy are increasing, it means that the model is getting better at fitting the training data while also generalizing well to new data.

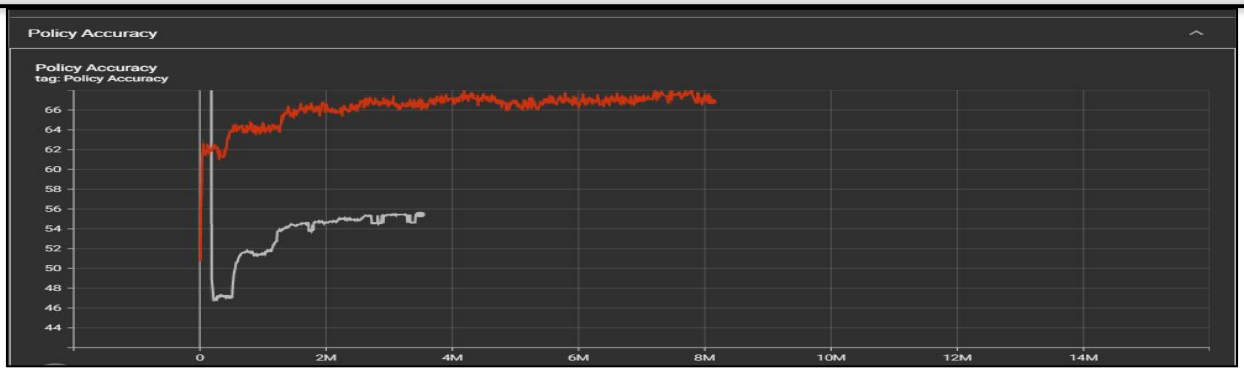


FIGURE 3. Policy Accuracy transition over 8 million epoch.

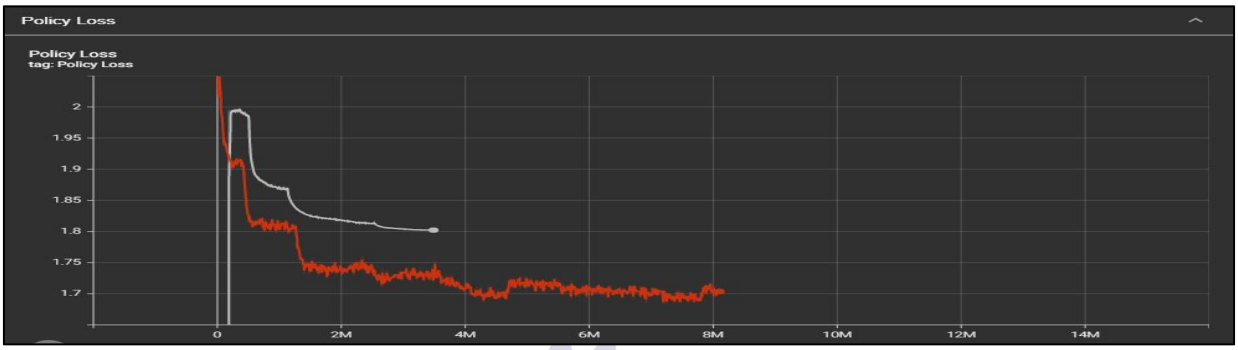


FIGURE 4. Policy Loss transition over 8 Million epoch.

Training and validation accuracy are increasing, it means that the model is getting better at fitting the training data while also generalizing well to new data.

XII. CONCLUSION:

To further protect against malware attacks on Android de- vices, it is important to regularly update the operating system and install security patches as they become available. This is especially important because these updates often include fixes for vulnerabilities that could be exploited by malware. In addition to keeping the operating system up to date, using a reputable antivirus software and only downloading apps from trusted sources can also help reduce the risk of infection. These measures can prevent malware from being installed on the device in the first place. In addition to these proactive measures, implementing strong passwords and enabling two-factor authentication can provide an extra layer of security for device access and online accounts. This can help to prevent unauthorized access to the device and the sensitive data it contains, even if malware is present.

It is essential for individuals and organizations to be proactive in maintaining the security of their Android devices, as the consequences of a successful malware attack can be severe. These can include loss of sensitive data, financial damage, and disruption of operations. Therefore, it is advisable to follow best practices and adopt a holistic approach to cybersecurity to ensure the protection of Android devices. It is essential for individuals and organizations to be proactive in maintaining the security of their Android devices, as the consequences of a successful malware attack can be severe. These can include loss of sensitive data, financial damage, and disruption of operations. Therefore, it is advisable to follow best practices and adopt a holistic approach to cybersecurity to ensure the protection of Android devices.

REFERENCE

[1]"Gartner Smartphone Reports," Gartner, 2019, https://www. gartner.com/en/newsroom/press-releases/2019 11-26- gartner-says-global-smartphone-demand-was-weak in-thi.

[2] W. Zhou, Y. Zhou, X. Jiang, and P. Ning, "Detecting repackaged smartphone applications in third-party android marketplaces, 2012 marketplaces," in Proceedings of the second ACM conference on Data and Application Security and Privacy, pp. 317–326, ACM, San Antonio, TX, USA, February 2012.

[3] R. Potharaju, M. Rahman, and B. Carbunar, "A longitudinal study of google play," IEEE Transactions on computational social systems, vol. 4, no. 3, pp. 135–149, 2017.

[4] "A Look at Google Bouncer." Trendlabs-Security Intelligence," 2019, https://blog.trendmicro.com/trendlabssecurity intelligence/a-look-at-google-bouncer/ accessed on.

[5] J. Oberheide and C. Miller, "Dissecting the android bouncer," SummerCon2012, vol. 95, p. 110, NewYork,2012.

[6] E. Cunningham, "Keeping You Safe with Google Play Protect," 2017, https://blog.google/products/android/ google play-protect/.

[7] T. Cho, H. Kim, and J. H. Yi, "Security assessment of code obfuscation based on dynamic monitoring in android things," IEEE Access, vol. 5, pp. 6361–6371, 2017.

[8] R. Binns, U. Lyngs, M. Van Kleek, J. Zhao, T. Libert, and N. Shadbolt, "(ird party tracking in the mobile ecosystem," in Proceedings of the 10th ACM Conference on Web Science, pp. 23–31, ACM, Amsterdam, Netherlands, May 2018.

[9] Z. Ma, H. Wang, Y. Guo, and X. Chen, "Libradar: fast and accurate detection of third-party libraries in android apps," in Proceedings of the 38th international conference on software engineering companion, pp. 653–656, ACM, Austin, TX, USA, May 2016.

[10] T. Isohara, K. Takemori, and A. Kubota, "Kernel based behavior analysis for android malware detection," in Proceedings of the 2011 Seventh International Conference on Computational Intelligence and Security, pp. 1011–1015, IEEE, Sanya, China, December 2011.

[11] M. Zheng, M. Sun, and J. C. Lui, "Droid analytics: a signature based analytic system to collect, extract, analyze and associate android malware," in Proceedings of the 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, pp. 163–171, IEEE, Melbourne, VIC, Australia, July 2013.

[12] C. Jarabek, D. Barrera, and J. Aycock, "(inav: truly lightweight mobile cloud-based anti-malware," in Proceedings of the 28th Annual Computer Security Applications Conference, pp. 209–218, ACM, Orlando, FL, USA, Decem ber 2012.

[13] W. Enck, D. Octeau, P. D. McDaniel, and S. Chaudhuri, "A study of android application security," USENIX security symposium, vol. 2, p. 2, 2011.

[14] E. Medvet and F. Mercaldo, "Exploring the usage of topic modeling for android malware static analysis," in Proceedings of the 2016 11th International Conference on Availability, Reliability and Security (ARES), pp. 609–617, IEEE, Salzburg, Austria, August 2016.

[15] T. Petsas, G. Voyatzis, E. Athanasopoulos, M. Poly chronakis, and S. Ioannidis, "Rage against the virtual ma chine: hindering dynamic analysis of android malware," in Proceedings of the 7th European Workshop on System Security, vol. 5, ACM, Amsterdam, Netherlands, April 2014.

[16] V. Kouliaridis, K. Barmpatsalou, G. Kambourakis, and S. Chen, "A survey on mobile malware detection techniques," IEICE- Transactions on Info and Systems, vol. E103.D, no. 2, pp. 204–211, 2020.

[17] S. K. Dash, G. Suarez-Tangil, S. Khan et al., "Droid scribe: classifying android malware based on runtime behavior," in Proceedings of the 2016 IEEE Security and Privacy Workshops (SPW), pp. 252–261, IEEE, San Jose, CA, USA, May 2016.

[18] N. Zhang, Y.-a. Tan, C. Yang, and Y. Li, "Deep learning feature exploration for android malware detection," Applied Soft Computing, vol. 102, Article ID 107069, 2021.

[19] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poor nachandran, and S. Venkatraman, "Robust intelligent mal ware detection using deep learning," IEEE Access, vol. 7, Article ID 46717, 2019.

[20] A. Alzubaidi, "Recent advances in android mobile malware detection: a systematic literature review," IEEE Access, vol. 9, 2021.

[21] G. Suarez-Tangil, S. K. Dash, M. Ahmadi, J. Kinder, G. Giacinto, and L. Cavallaro, "Droidsieve: fast and accurate classification of obfuscated android malware," in Proceedings of the 7th ACM on Conference on Data and Application Security and Privacy, pp. 309–320, Arizona, USE, March 2017.

[22] G. Suarez-Tangil, J. E. Tapiador, P. Peris-Lopez, and A. Ribagorda, "Evolution, detection and analysis of malware for smart devices," IEEE Communications Surveys Tutorials, vol. 16, no. 2, pp. 961–987, 2013.

[23] M. La Polla, F. Martinelli, and D. Sgandurra, "A sur vey on security for mobile devices," IEEE communications surveys tutorials, vol. 15, no. 1, pp. 446–471, 2012.

[24] P. Faruki, A. Bharmal, V. Laxmi et al., "Android security: a survey of issues, malware penetration, and defenses," IEEE communications surveys tutorials, vol. 17, no. 2, pp. 998– 1022, 2014.

[25] H. Cai, "Embracing mobile app evolution via continuous ecosystem mining and characterization," in Proceedings of the IEEE/ACM 7th International Conference on Mo bile Software Engineering and Systems, pp. 31–35, Madrid, Spain, July 2020.

[26] H. Cai, X. Fu, and A. Hamou-Lhadj, "A study of run time behavioral evolution of benign versus malicious apps in android," Information and Software Technology, vol. 122, Article ID 106291, 2020.

[27] F. Liu, H. Cai, G. Wang, D. Yao, K. O. Elish, and B. G. Ryder, "Mr-droid: a scalable and prioritized analysis of inter-app communication risks," in Proceedings of the 2017 IEEE Security and Privacy Workshops (SPW), pp. 189–198, IEEE, San Jose, CA, USA, May 2017.

[28] H. Cai, N. Meng, B. Ryder, and D. Yao, "Droidcat: effective android malware detection and categorization via app level profiling," IEEE Transactions on Information Forensics and Security, vol. 14, no. 6, pp. 1455–1470, 2018.

[29] X. Fu and H. Cai, "On the deterioration of learning based malware detectors for android," in Proceedings of the 2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), pp. 272-273, IEEE, Montreal, QC, Canada, May 2019.

[30] M. Ficco, "Malware analysis by combining multiple detectors and observation windows," IEEE Transactions on Computers, vol. 71, 2021