Malware Detection in Android Systems: UsingMachine Learning and Deep Learning

G. KANISHKA (PH.D)[1], P.AJITH[2], A.LIKITH SREE VAIBHAV[3], S.BHANU VARDHAN[4], G.REDDY BABU[5]

[1] *Asst Professor, Dept of CSE-CS, Madanapalle Institute of Technology & Science, Madanapalle, India*

[2, 3, 4, 5] *Dept of CSE-CS, Madanapalle Institute of Technology & Science, Madanapalle, India*

Abstract*— The increasing acceptance of the Android platform has led to a surge of malicious software, which poses an extreme risk to user confidentiality and device security. Traditional signature-based detection systems cannot keep up with the rapid expansion and sophistication of Android malware, which calls for more creative and adaptable protection measures. This paper investigates state-of-the art machine learning strategies for An- droid malware recognition. It discusses how various algorithms can be applied, how well they extract and learn from different feature sets, and how challenging it can be to deal with malware that is obfuscated and polymorphic. It also discusses malware developers adversarial tactics and the limitations of machine learning approaches in this never-ending arms race. To increase the precision of detection and response. We can observe numerous effective antimalware present in the internet whichcan effectively tackles the malware attacks and threats. So . I have examine some of them with their working algorithms from the internet to stop such attacks. According to our research from the internet, some deep learning techniques and android malware samples are essential for defending such malware attacks. Our motive is to find the hazardous malwares before installation of malware applications in android systems. This approach can detect whether the android application is infected or not, which reduces the severe risks and damages.

I. INTRODUCTION

During the daily scenario of mobile devices have highly engaged and predominant role why because of the infinite applications and services have been giving rise to us for using and utilizing the day to day life of mobile apps. The mobile will have modified the path of communication between the people and also the apps mainly think of installed on most of smart devices . Smart phone devices had re-fined sensors like camera, gyroscopes, and gravity sensors and microphones and also location sensors. These sensors can ultimately take the complete new innovation world of applications for the people and create a very large amount of information which was containing the data to highly complex size data. Mainly the android malware detection is to detect the vulnerability and also scan whether the app is safe or not without installation of any apps using chrome and any other browser. There are manymethodologies are used for largely not implemented for the identification of malware and malicious, but it is used to for risk assessment. And android operators physically enlarged the with the range of smart android devices. Majorly 90% in 2020 due to vulnerable sources distribution against the malware it can be risky and communicate with the system that notifies the people who are using the each application is installed in their phone. This System is lightly infective for personal. According to general we can't do differenced the permission based System. The methodologies are rapidly developed for the identifying the malware but it is used for the harm assessment.

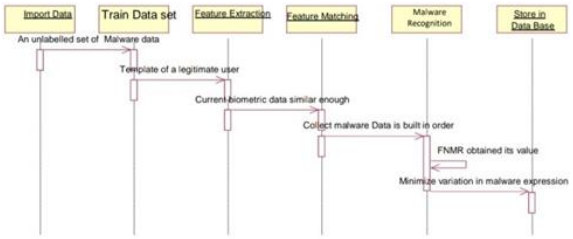

Figure 1: Architecture

II. BACKGROUND

The Bouncer software can scan all the third party apps which was installed by users with the certain period of time and also it

can scan only limited applications after that it will bypass all the applications. Mainly because of scanning phase it not to do anything. Bouncer there will be no malicious code scanned and transfer all the applications and skipped the certain ones at limited period of time and also the malicious app have higher chances of avoiding the malware and it required all the permissions for scanning the vulnerability and also it is lengthy process. The permission based methodologies which are required to are largely risk in assessing and used for detection and also detection system CNN and machine learning that are required to little preparation and pre- processing the order to acquire to the assembly process. In terms MalDozer has an easy-to-use design that requires little pre processing in order to acquire assembly processes. In terms of feature extraction, detection, and attribution, these are founded on concrete neural networks.

III. PROSPECTIVE WORK

Our suggested approach collects dual types of characteristics at android applications. That are mainly application programming interface and permissions and also it is comparable with the included files of xml and java files will be analyzed. Application programming interface that calls the function and it can taken from the java source files . whereas permissions are collected from manifest files. The gathered features will be integrated into an Android app the testing based deep learning model. deep learning is a model accurately differentiates between harmful and benign applications. The first is used to crawl malware from identified sources.

IV. LITERATURE SURVEY

A. MalDozer: an automatic system for detecting malware on Android using deep learning.

To keep up with the accelerated growth and sophistication of Android malware easy-going and intuitive security proposals are mandatory, the obstacle of common signature-based detection systems. This study mainly to think of cutting-edge machine learning descriptions for An- droid malware detection. It mainly tackles the purpose of the multiple algorithms, their how they are perform in extracting and learning from an extensive number of feature sets, and the difficulties of the battling polymorphic and obfuscated malware. It also think of the limitations of machine learning easy-going in this never-ending arms race, as well as the adversarial types adopted by malware authors. The study underlines how critical it is to include machine learning (ML) harms in the field where android devices gets destructed, mainly within the android security framework.

B. Droid detector: android malware characterization and detection using deep learning.

Droid detector is use by deep learning to identify and characterize malware on android. Many malwares are hidden on the huge variety of android security is jeopardized by seemingly benign apps in the market. Deep learning is a growing trend in artificial intelligence report and this study proposes combining static and dynamic scanning of applications through to identify malwares by using deep learning techniques or methods then, our deep learning based malwares to detect engine automatically detects dangerous apps. We tested droid detector on hundreds of android apps and scanned and reported. The findings show that deep learning is suitable for classifying malwares.

With additional training data, this strategy becomes even more effective. Droid detector has a detection accuracy of 96.76%, surpassing traditional machine learning methods.

An examination of 10 prominent antivirus softwares illustrates the need of improving our Android malware detection skills.

C. Droid Deep Learner: Droid Deep Learner uses deep learning to identify Android malware.

In android apps there are encounters with vicious apps by mobile stoners are on the increased as loop holes in android platform then the system are exploited by malwares authors to pierce particular information in the intentions, constantly using malwares to detect the security and integrity in order maintain the confidence in a various approaches have been studied in the field of malware discovery with more suitable at hiding for vicious intent through the use law and I becomes imperative for malware discovery ways to keep the changes in order to verify and detecting the changes through the machine learning techniques.

In this paper, we proposed a droid deep learner and android malware characterization with approach and identification by ensuring the required content of algorithms and malwares for comparing the being considerably together to discover the required experimental results has been approached.

V. METHODOLOGY

Summary of the selection criteria android deep flow is nothing but a deep learning model that classifies the exploits and crypt ware mainly based on data flowing in the android software applications. The deep flow which was a deep learning model is to building uses the network based module on deep learning to determine whether the application is attacked or not with some malicious. The salient features are retrieved at the application file. Deep data flow which can employs to the flow droid and examination static device, to category the complicated gathered information that goes down in applications.

To catalogue and sensitive files in android applications. Flow droid first checks from life duration and get back techniques and calls to resources and which can drown identify complicated the data flowing. Flow Droids approach the delicate information from the starting tip for stain research, focusing on to overcome the flow path.

A. Algorithm:

Discretization was utilized to discretize API method call sequences. The sequence identification must be compatible with our neural network. The solution is to express each identifier as a vector. They utilized one-hot vectors, with a value of one in the row and zero in the remainder. Each app generated vector sequences with unique API call instructions, each with a static size of K. Droid Delver exploited API method calls to access system resources and functionality in Android apps. To extract API calls, users must first unzip the apk file and access the class and the dex file. They utilized Aptos to extract api calls. This approach utilized the Most Relevant algorithm to pick API call sets. After extracting API calls, they classify them based on their proper procedure in the small code and place them in the API call block.

Algorithms	Accuracy
K-Nearest Neighbour	99.91 %
Support Vector Machine	99.89 %

Table 1: Algorithm Accuracy

B. Malware Detection Techniques:

APK Tool extracts APKs and decompiles dex files into smali code. The API call extractor extracts API calls from Smali code. The API call block generator in smali code will classify extracted APIs into blocks based on their appropriate approach. Each Android app will have an API call block. Deep Flow enables Flow Droid to process complicated data from several sources and sinks. Deep Flow uses the SUSI approach to classify extracted flows, capture features, and generate a feature vector. The DBN deep learning model uses feature vectors to classify the malware. The DBN The model features two crawler modules. The first crawler is used to identify the malware, while the second crawls innocent applications from the Google Play Store using another crawler.

C. Data Collection and Analysis:

During data collection and analysis, chosen publications were rigorously scrutinized for relevant information on An- droid malware detection using machine learning. To ensure. This review's findings were collected in an organized manner, ensuring consistency, completeness, and current information. We retrieved the following information from each paper: The study aimed to examine the dataset, machine learning approach, assessment metrics, and results. The data was evaluated to identify themes, trends, and gaps in the available research. This article provides an overview of machine learning algorithms for detecting Android malware, including its benefits and drawbacks. The findings of this analysis enabled the implementation of these solutions. The data from chosen publications was used to compare and contrast methodologies and identify topics for further investigation. This review provides a complete grasp of the field's present condition and identifies key hurdles and potential for future research. The data gathering and analysis procedure aimed to offer a complete review of the literature on using machine

learning to identify Android malware and guide future research in the field.

VI. RESULT AND ANALYSIS

This chapter will go over the ML(machine learning) studies which has been done for AMD(Android malware detection) systems. In addition, the first result displays the outcomes from MLP, J48, KNN and f-measure parameters were also used in this study to examine the various metrics of each classifier.

A. Results

The results indicates that KNN classifiers outperform other classifiers in terms of malware detection. The features chosen also had a significant impact on how well Android malware was detected. Figure 4 illustrates how the method may identify unknown malware with an accuracy rate of more than 99.91 %. This column represents the importance of this literature review on Malware discovery with Machine literacy.

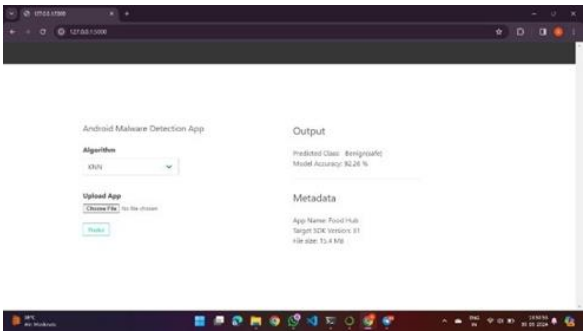

FIGURE 2: performance of classifier

These are the conclusions based on our final evaluation which of the papers chosen by using special care. There is a summary of the most important things we found from this review is presented below. Machine literacy is the mostly utilized in Malware detection. These machine literacy algorithms are used to identify the Android's Malware at the exact time of this study's review.

The algorithm implements that the machine literacy works accurately for the job. For the Android malware discovery, a different kind of ML(machine learning) algorithms are employed. colorful machine literacy algorithms, similar such as decision trees, neural networks, SVM(support vector machines), were used in the process of the program. This implements that, based on the particular system conditions and the behavior of the data being anatomized, There are many different algorithms that are very effective for this task. The Malware of android discovery that the performance of the system is dependent on the named datasets. A large variety of datasets such as real world and synthetic were used in reviewed data. The choosing of the dataset is pivotal to the performance of system and can significantly affect the eschewal comes. The compared studies have a vast range of evaluation criteria. The reviewed studies have a wide range of evaluation criteria .

This chapter will go over the machine learning studies that have been done for Android malware detection systems. In addition, the first result displays the outcomes from MLP,parameterswere also used in this study to examine the various metrics of each classifiers.

B. Comparative Analysis

This indicates that KNN classifiers outperform other classifiers in terms of malware detection. The features chosen also had a significant impact on how well Android malware was detected. Figure 2 illustrates how the method may identify unknown malware with an accuracy rate of more than 89 %.

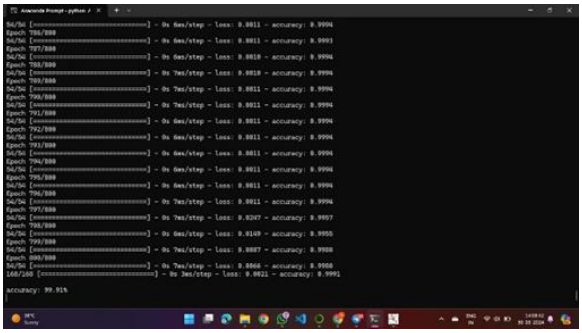

Figure 3: Percentage Accuracy

In the below figure, the horizontal axis displays the error detection rate and the vertical axis the detection rate. The ROC curve for each machine learning classifier was shownby five lines. Furthermore, because of the similarities under the same circumstances, ROC curves are difficult to compare. In the meantime, AUC was used to determine if the

detection accuracy was good or poor, as shown in the table below. As a result, an area of 1 represented a flawless prediction, whereas an area of 0.5 represented a poor prediction.

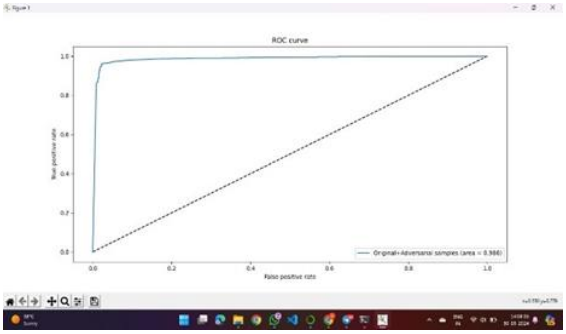

Figure 4: ROC Curve

CONCLUSION

This article covers several AMD(Android Malware Detection) Techniques utilizes DL(Deep Learning) Methods. Android's open nature can lead to malicious apps being hidden among seemingly benign apps in the store. These infections pose a significant danger to security of android .The unauthorized user can access user data such as mails, phone numbers, bank account numbers, and whereabouts. This review covers several AMD(android malware detection) techniques, including Droid Detector, MalDozer, Droid Deep Learner, and the Deep Flow. Mal-Dozer uses a CNN for android malware detection. It uses synchronous analysis and API calls to determine whether the program is contaminated with malware. The Droid Detector will detect using the Deep Belief Network. They utilized the We offer both static and dynamic analysis, including permissions, APIs, and dynamic behavior detection for the android malware. Using Droid Deep Learner uses Deep Belief Network for android malware detection.

REFERENCES

[1] E. B. Karbab, M. Debbabi, A. Derhab, and D. Mouheb, “MalDozer: Au- tomatic framework for android malware detection using deep learning,” Digital Investigation, vol. 24, 2018.

[2] Z. Yuan, Y. Lu, and Y. Xue, “Droiddetector: android malware char- acterization and detection using deep learning,” Tsinghua Science and Technology, vol. 21, no. 1, pp. 114–123, 2016.

[3] Z.Wang, J. Cai, S. Cheng, and W. Li, “DroidDeepLearner: Identifying Android malware using deep learning,” 2016 IEEE 37th Sarnoff Sym- posium, 2016.

[4] D. Zhu, H. Jin, Y. Yang, D. Wu, and W. Chen, “DeepFlow: Deep learning-based malware detection by mining Android application for abnormal usage of sensitive data,” 2017 IEEE Symposium on Computer and Communications (ISCC), 2017.

[5] X. Su, D. Zhang, W. Li, and K. Zhao, “A Deep Learning, Approach to Android Malware Feature Learning and Detection,” 2016 IEEE Trustcom/BigDataSE/ISPA, 2016.

[6] S. Hou, A. Saas, Y. Ye, and L. Chen, “DroidDelver: An Android Malware Detection System Using Deep Belief Network Based on API Call Blocks,” Web-Age Information Management Lecture Notes in Computer Science, pp. 54–66, 2016.

[7] R. Zachariah, K. Akash, M. S. Yousef, and A. M. Chacko, “Android malware detection a survey,” 2017 IEEE International Conference on Circuits and Systems (ICCS), 2017.

[8] P. Faruki, A. Bharmal, V. Laxmi, V. Ganmoor, M. S. Gaur, M. Conti, and M. Rajarajan, “Android Security: A Survey of Issues, Malware Penetration, and Defenses,” IEEE Communications Surveys Tutorials, vol. 17, no. 2, pp. 998–1022, 2015.

[9] A. I. A’Fifah, A. Ritahani, and A. Ahmad, “Comparative Performance of Deep Learning and Machine Learning Algorithms on Imbalanced Handwritten Data,” International Journal of Advanced Computer Science and Applications, vol. 9, no. 2, 2018.

[10] V. Rao, K. Hande, “A comparative study of static, dynamic and hybrid analysis techniques for android malware detection,” International Journal of Engineering Development and Research, pp. 1433-1436, 2017

[11] Li, L.; Li, D.; Bissyandé, T.F.; Klein, J.; Le Traon, Y.; Lo, D.; Cavallaro, L. Understanding android app piggybacking: A systematic study of malicious code grafting. IEEE Trans. Inf.

Forensics Secur. 2017, 12, 1269–1284. [CrossRef].

[12] Ashawa, M.A.; Morris, S. Analysis of Android malware detection techniques: A systematic review. Int. J. Cyber-Secur. Digit. Forensics 2019, 8, 177–187. [CrossRef]

[13] Suarez-Tangil, G.; Tapiador, J.E.; Peris-Lopez, P.; Ribagorda, A. Evolution, detection and analysis of malware for smart devices. IEEE Commun. Surv. Tutor. 2013, 16, 961–987. [CrossRef].

[14] Mos, A.; Chowdhury, M.M. Mobile Security: A Look into Android. In Proceedings of the 2020 IEEE International Conference on Electro Information Technology (EIT), Chicago, IL, USA, 31 July–1 August 2020; pp. 638–642. [CrossRef].

[15] Faruki, P.; Bharmal, A.; Laxmi, V.; Ganmoor, V.; Gaur, M.S.; Conti, M.; Rajarajan, M. Android security: A survey of issues, malware penetration, and defenses. IEEE Commun. Surv. Tutor. 2014, 17, 998–1022. [CrossRef].