International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 12, Issue 02, February 2023)

Review of Android Malware Prediction using Artificial Intelligence Techniques

Shivani Tiwari[1], Prof. Roopali Soni[2]

[1]*Research Scholar,* [2]*Associate Professor, Department of Computer Science Engineering, Oriental College of Technology, Bhopal, India*

***Abstract*— Android overlay is a feature that allows one program to draw over other applications by adding an additional View layer on top of the host View. Despite this, bad apps (malware) might take use of this feature to target users. Prior countermeasures focused on limiting the capabilities of overlays at the operating system level while sacrificing the usability of overlays to combat this threat; recently, the overlay mechanism has been substantially updated to prevent a variety of attacks; however, significant adversaries can still circumvent this protection. Malware is still a significant risk to computer security, which is why we need detection methods that rely on machine learning. While these detectors have a lot of potential, it is well recognized that they are susceptible to evasion assaults. The prediction of Android malware, along with performance improvements, is presented in this research.**

***Keywords*— Android, Malware, Artificial Intelligence, Secuiry, Attack, Cyber.**

I. INTRODUCTION

The explosion in the number of malware attacks against the Android platform may be attributed to the widespread use of the Android operating system in smartphones and other Internet-of-Things devices. Malware is a kind of software that poses a significant risk to the safety of computer systems and the services that are offered by such systems. For example, malware may steal personally identifiable information that is kept on mobile devices. Because of this effort, a stacking ensemble framework called SEDMDroid has been developed to detect malware on Android. To be more specific, it uses random feature subspaces and bootstrapping sampling approaches to produce subsets, and then it does Principal Component Analysis (PCA) on each of those subsets. This helps to guarantee that individuals have a wide range of characteristics. Keeping all of the primary components and using the whole dataset in the training of each base learner is how the correctness of the Multi-Layer Perception model is tested (MLP). The output of the ensemble members is used to learn the implicit extra information, and a Support Vector Machine (SVM) is used as the fusion classifier to provide the final prediction result [1].

Figure 1: Android malware

It is essential to search for malicious software on Android. Permission pair based detection systems are very promising for use in practical detection. There are many different detection schemes. Conventional methods, on the other hand, are unable to simultaneously satisfy criteria for practical application in terms of efficiency, intelligibility, and stability of detection performance. These requirements are intended to ensure that the method can be used effectively. Even while the most recent strategy is based on distinctions between common pairings of innocuous programs and malicious software, it is not stable enough to fulfill the requirements. This is due to the fact that modern malware has a tendency to demand additional rights in order to resemble innocuous applications, which renders the usage of the frequencies useless [3]. In recent years, machine learning (ML) has become an increasingly popular tool for the detection of malware across a variety of operating systems, including Android.

In order to stay up with the progression of malware, the detection models often need to be retrained on a regular basis (for example, once a month), using the data that is acquired out in the wild. This, however, opens the door to poisoning attacks, more precisely backdoor assaults, which are designed to thwart the learning process and provide evasion tunnels for maliciously altered copies of software. To this day, we have not been able to locate any previous study that investigated this very important issue in Android malware detection [4].

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 12, Issue 02, February 2023)

In recent years, ransomware has emerged as a significant danger that targets mobile devices, namely smartphones. A kind of malicious software known as ransomware locks down a mobile device's operating system and prohibits the owner of an infected device from accessing their data unless a ransom is paid. Attacks using ransomware have resulted in significant losses for persons and stakeholders all around the world.

Yet, owing to the continually evolving nature of ransomware features, the process of recognizing them has become much more difficult as ransomware families have proliferated at an alarming rate. Since they produce a large number of false positives, traditional malware detection approaches, such as statistically based preventive methods, are unable to battle the ever-evolving Ransomware. In point of fact, it is of utmost significance [6] to work on the development of a method that is not conventional and uses intelligence to protect against ransomware.

Deep learning's applicability to a variety of tasks has been made possible by the ready availability of large data sets and very inexpensive technology. With regard to safety, a number of initiatives have been undertaken to move the use of deep learning from the realm of image recognition or natural language processing into that of virus detection. In this research, we offer AdMat, a framework that is both simple and efficient, with the goal of characterizing Android apps by viewing them as photographs [8]. App marketplaces have evolved into a natural and accessible malware distribution route in recent years, despite the fact that they are an essential part of the modern mobile ecosystem. This is because they "lend legitimacy" to harmful programs. We have not yet seen an ML-based malware detection solution used at market sizes, despite the fact that machine learning (ML) methods have been intensively investigated for automated, robust malware detection over the last several years. We perform a joint research with T-Market, a major Android app store, which provides us with large-scale ground-truth data [9]. This allows us to systematically investigate the issues that are faced in the actual world. Malware designed for Android presents consumers with significant risks, which has led to an increased need for malware detection. While detecting Android malware in the cloud, privacy leaks and increased communication costs are typically unavoidable issues. As a result, the on-device Android malware detection will be the primary emphasis of this study. At the moment, on-device malware detectors are often taught on servers, and then the knowledge is transferred to mobile devices (e.g., smartphones). In actuality, training that takes place directly on the device is of utmost significance because of the necessity for offline upgrades.

On-device training, however, is difficult to accomplish on mobile devices due to the restricted resources available on mobile devices; this is particularly true for high-complexity malware detectors. We have developed a lightweight on-device Android malware detection that is based on the newly described wide learning algorithm [10].

II. Literature Survey

H. Zhu et al.,[1] show experimental results on two separate datasets collected by static analysis way to prove the effectiveness of the SEDMDroid. The first one extracts permission, sensitive API, monitoring system event and so on that are widely used in Android malwares as the features, and SEDMDroid achieves 89.07% accuracy in term of these multi-level static features. The second one, a public big dataset, extracts the sensitive data flow information as the features, and the average accuracy is 94.92%. Promising experiment results reveal that the proposed method is an effective way to identify Android malware.

A. Alzubaidi et al.,[2] In recent years, the global pervasiveness of smartphones has prompted the development of millions of free and commercially available applications. These applications allow users to perform various activities, such as communicating, gaming, and completing financial and educational tasks. These commonly used devices often store sensitive private information and, consequently, have been increasingly targeted by harmful malicious software. This paper focuses on the concepts and risks associated with malware, and reviews current approaches and mechanisms used to detect malware with respect to their methodology, associated datasets, and evaluation metrics.

H. Kato et al.,[3]. propose Android malware detection based on a Composition Ratio (CR) of permission pairs. We define the CR as a ratio of a permission pair to all pairs in an app. We focus on the fact that the CR tends to be small in malware because of unnecessary permissions. To obtain features without using the frequencies, we construct databases about the CR. For each app, we calculate similarity scores based on the databases. Finally, eight scores are fed into machine learning (ML) based classifiers as features. By doing this, stable performance can be achieved. Since our features are just eight-dimensional, the proposed scheme takes less training time and is compatible with other ML based schemes. Furthermore, our features can quantitatively offer clear information that helps human to understand detection results. Our scheme is suitable for practical use because all the requirements can be met. By using real datasets, our results show that our scheme can detect malware with up to 97.3% accuracy.

Besides, compared with an existing scheme, our scheme can reduce the feature dimensions by about 99% with maintaining comparable accuracy on recent datasets.

C. Li et al et al.,[4] motivated to study the backdoor attack against Android malware detectors. The backdoor is created and injected into the model stealthily without access to the training data and activated when an app with the trigger is presented. We demonstrate the proposed attack on four typical malware detectors that have been widely discussed in academia. Our evaluation shows that the proposed backdoor attack achieves up to 99% evasion rate over 750 malware samples. Moreover, the above successful attack is realised by a small size of triggers (only four features) and a very low data poisoning rate (0.3%).

L. Gong, Z. Li et al.,[5] To address these shortcomings, a more pragmatic approach is to enable early detection of overlay-based malware during the app market review process, so that all the capabilities of overlays can stay unchanged. For this purpose, in this paper we first conduct a large-scale comparative study of overlay characteristics in benign and malicious apps, and then implement the OverlayChecker system to automatically detect overlay-based malware for one of the worlds largest Android app stores. In particular, we have made systematic efforts in feature engineering, UI exploration, emulation architecture, and run-time environment, thus maintaining high detection accuracy (97% precision and 97% recall) and short per-app scan time (1.7 minutes) with only two commodity servers, under an intensive workload of 10K newly submitted apps per day.

I. Almomani et al.et al.,[6] introduces a new methodology for the detection of Ransomware that is depending on an evolutionary-based machine learning approach. The binary particle swarm optimization algorithm is utilized for tuning the hyperparameters of the classification algorithm, as well as performing feature selection. The support vector machines (SVM) algorithm is used alongside the synthetic minority oversampling technique (SMOTE) for classification. The utilized dataset is collected from various sources, which consists of 10,153 Android applications, where 500 of them are Ransomware. The performance of the proposed approach SMOTE-tBPSO-SVM achieved merits over traditional machine learning algorithms by having the highest scores in terms of sensitivity, specificity, and g-mean.

F. Mercaldo and A. Santone et al.,[7] Several techniques to overcome the weaknesses of the current signature based detection approaches adopted by free and commercial anti-malware were proposed by industrial and research communities.

These techniques are mainly supervised machine learning based, requiring optimal class balance to generate good predictive models. In this paper, we propose a method to infer mobile application maliciousness by detecting the belonging family, exploiting formal equivalence checking. We introduce a set of heuristics to reduce the number of mobile application comparisons and we define a metric reflecting the application maliciousness. Real-world experiments on 35 Android malware families (ranging from 2010 to 2018) confirm the effectiveness of the proposed method in mobile malware detection and family identification.

L. N. Vu and S. Jung, "AdMat et al.,[8] The novelty of our study lies in the construction of an adjacency matrix for each application. These matrices act as “input images” to the Convolutional Neural Network model, allowing it to learn to differentiate benign and malicious apps, as well as malware families. During the experiment, we found that AdMat was able to adapt to a variety of training ratios and achieve the average detection rate of 98.26% in different malware datasets. In classification tasks, it also successfully recognized over 97.00% of different malware families with limited number of training data.

L. Gong et al et al.,[9] Our study illustrates that the key to successfully developing such systems is multifold, including feature selection and encoding, feature engineering and exposure, app analysis speed and efficacy, developer and user engagement, as well as ML model evolution. Failure in any of the above aspects could lead to the “wooden barrel effect” of the whole system. This article presents our judicious design choices and first-hand deployment experiences in building a practical ML-powered malware detection system. It has been operational at T-Market, using a single commodity server to check ~12K apps every day, and has achieved an overall precision of 98.9 percent and recall of 98.1 percent with an average per-app scan time of 0.9 minutes.

W. Yuan, Y. Jiang et al.,[10] Our detector mainly uses one-shot computation for model training. Hence it can be fully or incrementally trained directly on mobile devices. As far as detection accuracy is concerned, our detector outperforms the shallow learning-based models, including support vector machine (SVM) and AdaBoost, and approaches the deep learning-based models multilayer perceptron (MLP) and convolutional neural network (CNN). Moreover, our detector is more robust to adversarial examples than the existing detectors, and its robustness can be further improved through on-device model retraining. Finally, its advantages are confirmed by extensive experiments, and its practicality is demonstrated through runtime evaluation on smartphones.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 12, Issue 02, February 2023)

K. Liu et al.,[11] presents complements the previous reviews by surveying a wider range of aspects of the topic. This paper presents a comprehensive survey of Android malware detection approaches based on machine learning. We briefly introduce some background on Android applications, including the Android system architecture, security mechanisms, and classification of Android malware. Then, taking machine learning as the focus, we analyze and summarize the research status from key perspectives such as sample acquisition, data preprocessing, feature selection, machine learning models, algorithms, and the evaluation of detection effectiveness. Finally, we assess the future prospects for research into Android malware detection based on machine learning. This review will help academics gain a full picture of Android malware detection based on machine learning. It could then serve as a basis for subsequent researchers to start new work and help to guide research in the field more generally.

D. Li and Q. Li et al.,[12] Ensemble learning typically facilitates countermeasures, while attackers can leverage this technique to improve attack effectiveness as well. This motivates us to investigate which kind of robustness the ensemble defense or effectiveness the ensemble attack can achieve, particularly when they combat with each other. We thus propose a new attack approach, named mixture of attacks, by rendering attackers capable of multiple generative methods and multiple manipulation sets, to perturb a malware example without ruining its malicious functionality. This naturally leads to a new instantiation of adversarial training, which is further geared to enhancing the ensemble of deep neural networks. We evaluate defenses using Android malware detectors against 26 different attacks upon two practical datasets. Experimental results show that the new adversarial training significantly enhances the robustness of deep neural networks against a wide range of attacks; ensemble methods promote the robustness when base classifiers are robust enough, and yet ensemble attacks can evade the enhanced malware detectors effectively, even notably downgrading the VirusTotal service.

III. Challenges

Android malware prediction using artificial intelligence (AI) techniques faces several challenges. Some of the significant challenges are:

1. *Lack of labeled data:* One of the significant challenges is the lack of labeled data required to train machine learning algorithms. It is difficult to obtain a large dataset of labeled samples for different types of malware.
2. *Feature engineering:* Extracting useful features from the raw data is essential for building accurate machine learning models. However, feature engineering for Android malware prediction is challenging due to the dynamic nature of mobile applications and the diversity of Android devices.
3. *Class imbalance:* The number of samples in the malware class is usually much smaller than that in the benign class, which can lead to a class imbalance problem. This can affect the accuracy of the machine learning models, and can also result in a biased classifier.
4. *Adversarial attacks:* Attackers can use various techniques to evade detection by AI-based malware detectors. For example, they can use obfuscation techniques or encrypt the malicious code to make it difficult to detect.
5. *Performance overhead:* Machine learning algorithms require significant computational resources and can impose significant overhead on mobile devices, which are typically resource-constrained.
6. *Privacy concerns:* Android malware prediction using AI techniques requires access to sensitive user data, such as the list of installed applications and network traffic. This can raise privacy concerns and can lead to legal and ethical issues.

IV. Conclusion

Android apps are evolving quickly throughout the mobile ecosystem, but at the same time, a never-ending flood of malicious Android software is also appearing. A multitude of researchers have examined the issue of detecting malware on Android devices and have proposed several hypotheses and approaches, each coming from a unique point of view. The research that has been done so far reveals that using machine learning to identify Android malware is a method that is both successful and promising. Despite this, there are evaluations that have investigated a variety of concerns about Android malware detection based on machine learning. In the future, develop a prediction model with a higher degree of accuracy by making use of an effective machine learning classification strategy.

REFERENCES

[1] H. Zhu, Y. Li, R. Li, J. Li, Z. You and H. Song, "SEDMDroid: An Enhanced Stacking Ensemble Framework for Android Malware Detection," in IEEE Transactions on Network Science and Engineering, vol. 8, no. 2, pp. 984-994, 1 April-June 2021, doi: 10.1109/TNSE.2020.2996379.

[2] A. Alzubaidi, "Recent Advances in Android Mobile Malware Detection: A Systematic Literature Review," in IEEE Access, vol. 9, pp. 146318-146349, 2021, doi: 10.1109/ACCESS.2021.3123187.

[3] H. Kato, T. Sasaki and I. Sasase, "Android Malware Detection Based on Composition Ratio of Permission Pairs," in IEEE Access, vol. 9, pp. 130006-130019, 2021, doi: 10.1109/ACCESS.2021.3113711.

[4] C. Li et al., "Backdoor Attack on Machine Learning Based Android Malware Detectors," in IEEE Transactions on Dependable and Secure Computing, doi: 10.1109/TDSC.2021.3094824.

[5] L. Gong, Z. Li, H. Wang, H. Lin, X. Ma and Y. Liu, "Overlay-based Android Malware Detection at Market Scales: Systematically Adapting to the New Technological Landscape," in IEEE Transactions on Mobile Computing, doi: 10.1109/TMC.2021.3079433.

[6] I. Almomani et al., "Android Ransomware Detection Based on a Hybrid Evolutionary Approach in the Context of Highly Imbalanced Data," in IEEE Access, vol. 9, pp. 57674-57691, 2021, doi: 10.1109/ACCESS.2021.3071450.

[7] F. Mercaldo and A. Santone, "Formal Equivalence Checking for Mobile Malware Detection and Family Classification," in IEEE Transactions on Software Engineering, doi: 10.1109/TSE.2021.3067061.

[8] L. N. Vu and S. Jung, "AdMat: A CNN-on-Matrix Approach to Android Malware Detection and Classification," in IEEE Access, vol. 9, pp. 39680-39694, 2021, doi: 10.1109/ACCESS.2021.3063748.

[9] L. Gong et al., "Systematically Landing Machine Learning onto Market-Scale Mobile Malware Detection," in IEEE Transactions on Parallel and Distributed Systems, vol. 32, no. 7, pp. 1615-1628, 1 July 2021, doi: 10.1109/TPDS.2020.3046092.

[10] W. Yuan, Y. Jiang, H. Li and M. Cai, "A Lightweight On-Device Detection Method for Android Malware," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 51, no. 9, pp. 5600-5611, Sept. 2021, doi: 10.1109/TSMC.2019.2958382.

[11] K. Liu, S. Xu, G. Xu, M. Zhang, D. Sun and H. Liu, "A Review of Android Malware Detection Approaches Based on Machine Learning," in IEEE Access, vol. 8, pp. 124579-124607, 2020, doi: 10.1109/ACCESS.2020.3006143.

[12] D. Li and Q. Li, "Adversarial Deep Ensemble: Evasion Attacks and Defenses for Malware Detection," in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 3886-3900, 2020, doi: 10.1109/TIFS.2020.3003571.