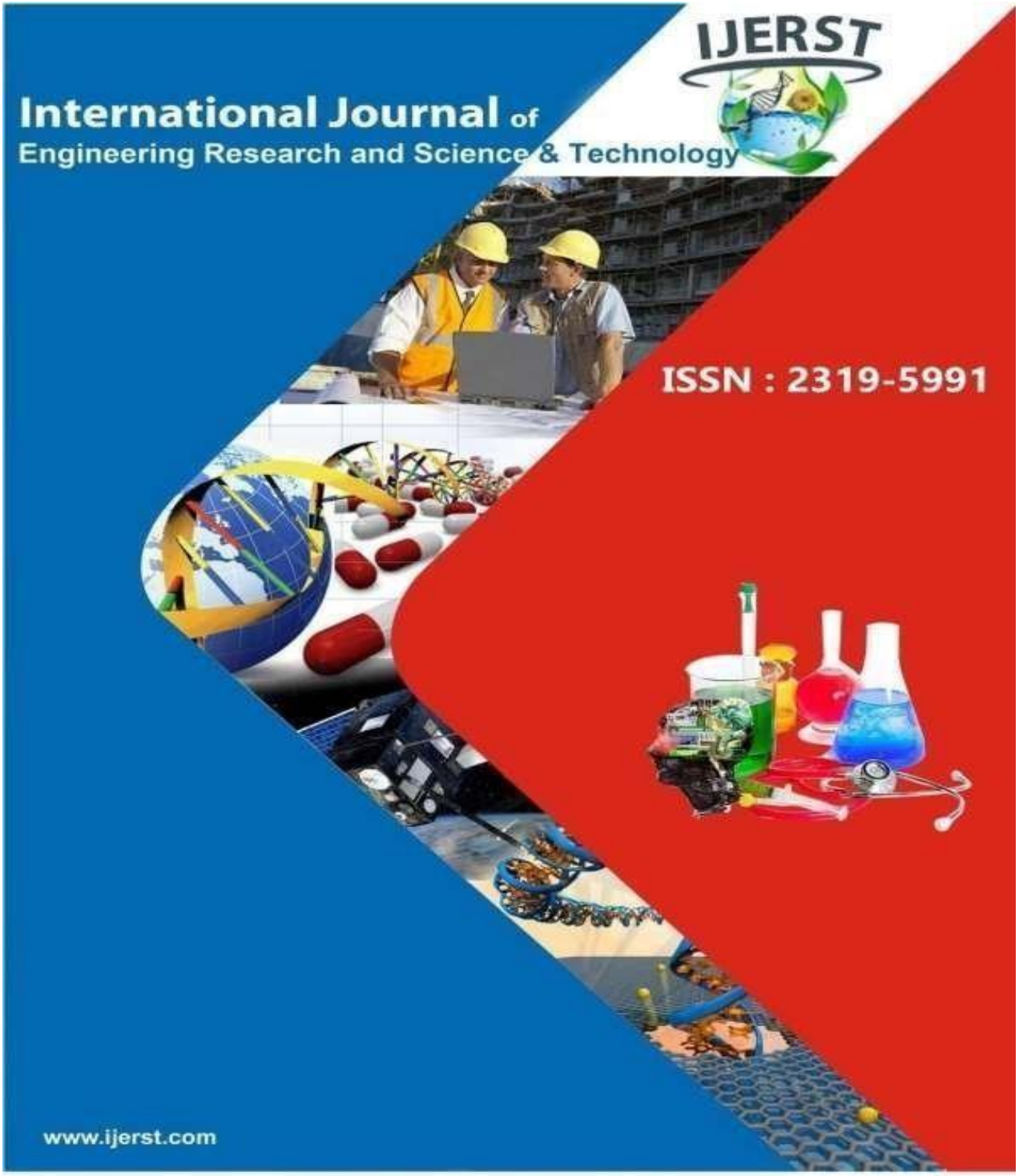


Email: editor@ijerst.com or editor.ijerst@gmail.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 20, Issue 2, 2024

MALWARE DETECTION A FRAME WORK FOR REVERSE ENGINEERED ANDROID APPLICATIONS THROUGH MACHINE LEARNING ALGORITHMS

[1]MR.RAMA BHADRA RAO MADDU, [2]ELIPAY MAHIM KUMAR

[1](Assistant Professor), MCA, Swarnandhra College

[2]MCA, scholar, Swarnandhra College

ABSTRACT

Today, Android is one of the most used operating systems in smartphone technology. This is the main reason, Android has become the favorite target for hackers and attackers. Malicious codes are being embedded in Android applications in such a sophisticated manner that detecting and identifying an application as a malware has become the toughest job for security providers. In terms of ingenuity and cognition, Android malware has progressed to the point where they're more impervious to conventional detection techniques. Approaches based on machine learning have emerged as a much more effective way to tackle the intricacy and originality of developing Android threats. They function by first identifying current patterns of malware activity and then using this information to distinguish between identified threats and unidentified threats with unknown behavior. This research paper uses Reverse Engineered Android applications' features and Machine Learning algorithms to find vulnerabilities present in Smartphone applications. Our contribution is twofold. Firstly, we propose a model that incorporates more innovative static feature sets with the largest current datasets of malware samples than conventional methods. Secondly, we have used ensemble learning with machine learning algorithms such as AdaBoost, SVM, etc. to improve our model's performance. Our experimental results and findings exhibit 96.24% accuracy to detect extracted malware

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 20, Issue 2, 2024

from Android applications, with a 0.3 False Positive Rate (FPR). The proposed model incorporates ignored detrimental features such as permissions, intents, API calls, and so on, trained by feeding a solitary arbitrary feature, extracted by reverse engineering as an input to the machine.

1.INTRODUCTION

This degree ensures that mobile devices play a crucial role in the majority of people's everyday activities. The majority of mobile devices are now Android-powered. In fact, during the last few years, Android smartphones have captured an average of 80% of the worldwide market share. Malware specifically designed to infect Android devices has also grown in recent years, paralleling Android's expansion into more and more smart phones and users worldwide. The risk it presents is high since it is an open-source OS and malicious programmers and writers may install malicious features, permissions, and components into Android applications. Also, it has the possibility to integrate with third-party software, which may make it more versatile, but it also opens it up to assaults from rogue devices. An individual's privacy and the safety of their valuables are more at risk as the number of applications for mobile devices grows. Since then, app security has declined, leading to incidents like data theft, SMS fraud, ransomware, etc.
Unlike static analysis methods, which involve manually examining files like AndroidManifest.xml, source code, and Dalvik Byte Code, as well as analyzing managed environments to understand how they handle programs, Machine Learning involves learning the basic rules and habits of both good and bad app settings and then data-venabling. A simple solution to the problem of manually extracting static features from reverse-engineered Android applications and then training a model to predict whether or not these apps contain malware is to use the support vector machine (SVM) algorithm, logistic progression, ensemble learning, or any of a number of other machine learning techniques..

2.LITERATURE SURVEY

(Gavrilov Dragot and Cimpoesu Mihai) Using Machine Learning for Malware Detection This is the collection of papers presented at the international conference on computer science and IT.
We provide a flexible framework that can be

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 20, Issue 2, 2024

used with various machine learning algorithms to effectively differentiate between clean and malicious files, with the goal of reducing the amount of false positives. The study lays forth the concepts of our framework, which is based on working with cascade one-sided perceptrons and cascade kernelized one-sided perceptrons. We were able to successfully test our framework on medium-size datasets of malware and clean files, and then we scaled up the principles underlying it such that we can deal with really huge datasets. [2] in Analysis and Detection of Malware with the Use of Machine Learning Algorithms Muhammad Shoaib Akthar and Tao Feng 2022 Symmetry at Lanzhou University of Technology....

this link: https://doi.org/10.3390/sym14112304

" We used a plethora of machine learning algorithms to detect harmful threats and viruses. The method with the highest detection ratio was chosen for system utilization because of its great accuracy. A benefit of the confusion matrix was that it counted the amount of false positives and false negatives, which gave more information about the system's performance. In particular, it was shown that by utilizing the results of malware analysis and detection with machine learning algorithms to calculate the difference in correlation symmetry integrals (Naive Byes, SVM, J48, RF, and with the proposed approach), it was possible to detect harmful traffic on computer systems, and thus improve the security of Type equation here.computer networks. Findings shown that DT(99%), CNN(98.76%), and SVM(96.41%) outperformed competing classifiers in terms of detection accuracy. In a specific dataset, we examined the malware detection results of DT, CNN, and SVM algorithms on a tiny FPR. DT achieved 2.01%, CNN achieved 3.97%, and SVM achieved 4.63% (). Given the prevalence and sophistication of malicious software, these findings are noteworthy. the third Malware Detection by Framework Reverse Engineering of Android Applications using Machine Algorithms, Beenish Urooj and Munam Ali Shah 2022 Presats University is this place.

Hackers and attackers now love targeting Android. The level of sophistication in malicious code embedding in Android apps has made it the most challenging task for security companies to detect and classify applications as malware. Android malware

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 20, Issue 2, 2024

has become more intelligent and sophisticated to the point that it may evade traditional detection methods. The complexity and novelty of new Android threats have made machine learning-based approaches the way to go. They work by observing patterns of malware activity in the present and then utilizing that data to differentiate between known dangers and novel threats whose behavior is unknown. In order to identify security flaws in smartphone apps, this study employs Machine Learning algorithms and the properties of reverse-engineered Android apps. We bring two things to the table. We begin by presenting a model that, in comparison to the status quo, makes use of the biggest existing datasets of malware samples together with more novel static feature sets. The second thing we did to make our model better was to apply ensemble learning using several machine learning methods, such as AdaBoost and Support Vector Machine (SVM). In terms of detecting malware extracted from Android apps, our trial results and findings demonstrate an accuracy of 96.24%, with a False Positive Rate (FPR) of 0.3.

3. EXISTING SYSTEM

This linked paper proposes strategies that improve malware detection predictions and address important problems. There has been a lot of study on improving detection rate efficiency; some studies have concentrated on raising accuracy, some on giving a bigger dataset, yet others on implementing these methods using other feature sets, and still more on combining all of these. With the goal of better organizing the Android Market, the authors of [21] provide a method for identifying malicious applications for Android devices. The suggested framework's end goal is to equip Android users with a malware detection solution that uses machine learning to weed out malicious applications and safeguard their personal information. In order to identify malicious or benign Android applications, this system keeps an eye on various permission-based features and events, and then uses machine learning

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 20, Issue 2, 2024

classifiers to analyze these properties.

Drawbacks

The system does not have any machine learning algorithms or assembly learning modules.

The features of reverse-engineered applications have not been included into the system.

NewApproach

Using about 56,000 characteristics from these categories, we provide a new subset of features—consisting of seven more feature sets—for static detection of Android malware. We evaluate their robustness using a dataset consisting of over 500,000 Android apps, both safe and harmful, and the largest collection of malware samples compared to any state-of-the-art method. With just 0.3% of findings being false positives, the results achieve a detection accuracy boost of 96.24%.

2) A Boosting ensemble learning strategy (AdaBoost) with a Decision Tree based on the binary classification has been used, along with six classifier models or machine learning algorithms, to improve our prediction rate, thanks to the extra characteristics. 3) Compared to state-of-the-art methods, our model is trained on the most up-to-date malware samples gathered in the last several years, including those at the most current Android API level.

The Benefits

The suggested approach selects the attributes according to their capacity to show all datasets. An efficient method for selecting functions is introduced, which improves efficiency by decreasing the size of the dataset and the amount of time lost during classification.

A bigger collection of features is also employed for categorization in the method used in this investigation. Despite how common this issue is in ML, using the right

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 20, Issue 2, 2024

model for detection or classification may have a major effect on the data's high dimensionality.

.4. OUTPUT SCREENS

User Register:



User Login:



Admin Login:



Browse Apps Dataset Train &Test:



View All Remote Users:



5. CONCLUSION

We developed a system that can identify Android apps that are malicious as part of our study. By using many ML

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 20, Issue 2, 2024

components, the suggested method successfully identifies malicious Android apps with a 96.24% success rate. In order to train the model with both benign and malicious datasets, we use python build modules and split shuffle functions. We begin by defining and selecting functions to record and analyze the activity of Android applications. We next use reverse application engineering and AndroGuard to extract features into binary vectors. Using improved and bigger feature and sample sets, our experimental results demonstrate that our proposed model achieves 96% accuracy in the provided context. When compared, the results from ensemble and strong learner methods are far superior..

6.REFERENCES

[1] A. O. Christiana, B. A. Gyunka, and A. Noah, "Android Malware Detection through Machine Learning Techniques: A Review," Int. J. Online Biomed. Eng. IJOE, vol. 16, no. 02, p. 14, Feb. 2020, doi: 10.3991/ijoe.v16i02.11549. [2] D. Ghimire and J. Lee, "Geometric Feature-Based Facial Expression Recognition in Image Sequences Using Multi-Class AdaBoost and Support Vector Machines," Sensors, vol. 13, no. 6, pp. 7714–7734, Jun. 2013, doi: 10.3390/s130607714. [3] R. Wang, "AdaBoost for Feature Selection, Classification and Its Relation with SVM, A Review," Phys. Procedia, vol. 25, pp. 800–807, 2012, doi: 10.1016/j.phpro.2012.03.160. [4] J. Sun, H. Fujita, P. Chen, and H. Li, "Dynamic financial distress prediction with concept drift based on time weighting combined with Adaboost support vector machine ensemble," Knowl.-Based Syst., vol. 120, pp. 4–14, Mar. 2017, doi: 10.1016/j.knosys.2016.12.019. [5] A. Garg and K. Tai, "Comparison of statistical and machine learning methods in modelling of data with multicollinearity," Int. J. Model. Identif. Control, vol. 18, no. 4, p. 295, 2013, doi: 10.1504/IJMIC.2013.053535. [6] C. P. Obite, N. P. Olewuezi, G. U. Ugwuanyim, and D. C. Bartholomew, "Multicollinearity Effect in Regression Analysis: A Feed Forward Artificial Neural Network Approach," Asian J. Probab. Stat., pp. 22–33, Jan. 2020, doi: 10.9734/ajpas/2020/v6i130151. [7] W. Wang et al., "Constructing Features for Detecting Android Malicious Applications: Issues, Taxonomy and Directions," IEEE Access, vol. 7, pp. 67602–67631, 2019, doi: 10.1109/ACCESS.2019.2918139. [8] B.

ISSN 2319-5991 www.ijerst.com
Vol. 20, Issue 2, 2024

Rashidi, C. Fung, and E. Bertino, "Android malicious application detection using support vector machine and active learning," in 2017 13th International Conference on Network and Service Management (CNSM), Tokyo, Nov. 2017, pp. 1–9. doi: 10.23919/CNSM.2017.8256035. [9] J. Li, L. Sun, Q. Yan, Z. Li, W. Srisa-an, and H. Ye, "Significant Permission Identification for Machine-Learning-Based Android Malware Detection," IEEE Trans. Ind. Inform., vol. 14, no. 7, pp. 3216–3225, Jul. 2018, doi: 10.1109/TII.2017.2789219. [10] G. Suarez-Tangil, J. E. Tapiador, P. Peris-Lopez, and J. Blasco, "Dendroid: A text mining approach to analyzing and classifying code structures in Android malware families," Expert Syst. Appl., vol. 41, no. 4, pp. 1104–1117, Mar. 2014, doi: 10.1016/j.eswa.2013.07.106. [11] M. Magdum, "Permission based Mobile Malware Detection System using Machine Learning Techniques," vol. 14, no. 6, pp. 6170–6174, 2015. [12] M. Qiao, A. H. Sung, and Q. Liu, "Merging Permission and API Features for Android Malware Detection," in 2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI), Kumamoto, Japan, Jul. 2016, pp. 566–571. doi: 10.1109/IIAI- AAI.2016.237. [13] D. O. Sahin, O. E. Kural, S. Akleylek, and E. Kilic, "New results on permission based static analysis for Android malware," in 2018 6th International Symposium on Digital Forensic and Security (ISDFS), Antalya, Mar. 2018, pp. 1–4. doi: 10.1109/ISDFS.2018.8355377. [14] A. Mahindru and A. L. Sangal, "MLDroid—framework for Android malware detection using machine learning techniques," Neural Comput. Appl., vol. 33, no. 10, pp. 5183–5240, May 2021, doi: 10.1007/s00521-020-05309-4. [15] X. Su, D. Zhang, W. Li, and K. Zhao, "A Deep Learning Approach to Android Malware Feature Learning and Detection," in 2016 IEEE Trustcom/BigDataSE/ISPA, Tianjin, China, Aug. 2016, pp. 244–251. doi: 10.1109/TrustCom.2016.0070. [16] K. A. Talha, D. I. Alper, and C. Aydin, "APK Auditor: Permission-based Android malware detection system," Digit. Investig., vol. 13, pp. 1–14, Jun. 2015, doi: 10.1016/j.diin.2015.01.001. [17] A. Mahindru and P. Singh, "Dynamic Permissions based Android Malware Detection using Machine Learning Techniques," in Proceedings of the 10th Innovations in Software Engineering Conference, Jaipur India, Feb. 2017, pp. 202–210. doi: 10.1145/3021460.3021485. [18] U. Pehlivan, N. Baltaci, C. Acarturk, and N.

ISSN 2319-5991 www.ijerst.com
Vol. 20, Issue 2, 2024

Baykal, "The analysis of feature selection methods and classification algorithms in permission based Android malware detection," in 2014 IEEE Symposium on Computational Intelligence in Cyber Security (CICS), Orlando, FL, USA, Dec. 2014, pp. 1–8. doi: 10.1109/CICYBS.2014.7013371. [19] M. Kedziora, P. Gawin, M. Szczepanik, and I. Jozwiak, "Malware Detection Using Machine Learning Algorithms and Reverse Engineering of Android Java Code," Int. J. Netw. Secur. Its Appl., vol. 11, no. 01, pp. 01–14, Jan. 2019, doi: 10.5121/ijnsa.2019.11101. [20] X. Liu and J. Liu, "A Two-Layered Permission-Based Android Malware Detection Scheme," in 2014 2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, Oxford, United Kingdom, Apr. 2014, pp. 142–148. doi: 10.1109/MobileCloud.2014.22. [21] "Permission-Based Android Malware Detection | Semantic Scholar." https://www.semanticscholar.org/paper/Permission-Based-Android- Malware-Detection-Aung-Zaw/c8576b5df33813fe8938cbb19e35217ee21fc80b (accessed Oct. 31, 2021). [22] D. Arp, M. Spreitzenbarth, M. Hübner, H. Gascon, and K. Rieck, "Drebin: Effective and Explainable Detection of Android Malware in Your Pocket," presented at the Network and Distributed System Security Symposium, San Diego, CA, 2014. doi: 10.14722/ndss.2014.23247. [23] H. Cai, N. Meng, B. Ryder, and D. Yao, "DroidCat: Effective Android Malware Detection and Categorization via App-Level Profiling," IEEE Trans. Inf. Forensics Secur., vol. 14, no. 6, pp. 1455–1470, Jun. 2019, doi: 10.1109/TIFS.2018.2879302. [24] P. Rovelli and Ý. Vigfússon, "PMDS: Permission-Based Malware Detection System," in Information Systems Security, vol. 8880, A. Prakash and R. Shyamasundar, Eds. Cham: Springer International Publishing, 2014, pp. 338–357. doi: 10.1007/978-3-319-13841-1_19. [25] M. S. Alam and S. T. Vuong, "Random Forest Classification for Detecting Android Malware," in 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, Beijing, China, Aug. 2013, pp. 663–669. doi: 10.1109/GreenCom-iThings- CPSCom.2013.122.