A novel approach for Android Malware Detection using Machine learning techniques

Mangalapilly Krishna Prasad [1], Dr.P.Chiranjeevi[2]

Student[1], Professor[2]

Amrita Sai Institute of Science and Technology Paritala-521180

Autonomous NAAC with A Grade, Andhra Pradesh, India.

Abstract -The presence of malware presents a major problem in both the operating system and software domains. The Android operating system is also included with these problems. The Android operating system is also included with these problems. Compared to other operating systems Android is one of the most used operating system in smartphones with a unique 2 billion users and has 74% market [1]. Other approaches, utilizing signature-based methods, have been employed for the detection of malware. Despite their application, these techniques were unable to detect unknown malware effectively. Even with the availability of various detection and analysis techniques, the issue of accurately detecting new malware remains a critical issue

In a survey conducted in 2006 by Microsoft company they have found 45000 variants of malwares like Trojan, backdoor and bots [9]. The focus of our research is to investigate and underscore the prevailing approaches employed in identifying and analysing malevolent code specifically designed for Android platforms. Alongside our study, we suggest the implementation of machine-learning algorithms for the analysis of such malware, complemented by semantic analysis methodologies. Throughthe use of ML, many procedures can be executed on interconnected data which involves classification, regression and clustering. ML algorithms are being used inmalware detection techniques since many years [2]. In this paper we focus on a new android malware detection method which uses GUI.

I. INTRODUCTION

Many survey reports have shown that nearly I million malware files are being evolved every day and these malware files have become a step to many cybercrimes which would affect the world's economy. [3]. The key intention of malware is to impede, pocket or do some violations. Malware possesses the power to invade any kind of machine processing application. The detection techniques used for smartphones are trailing in comparison with the rapid rise of mobile user base. Android is themost widely used mobile operating system (OS). As of February 2023, its market share was72.26%. Based on the McAfee mobile threat report, there is a vast upsurge in backdoors, fake applications and banking Trojans for mobile devices [4]. The Google android market also offers no certainty that all the applications included are threat free. There have been reports indicating the presence of Trojan applications that, when downloaded, secretly implant malicious code into devices. Android threats include SMS-Based Attacks, App Store Policy Violations, Phishing Attacks, bots. There are around 3.6 billion android users worldwide.

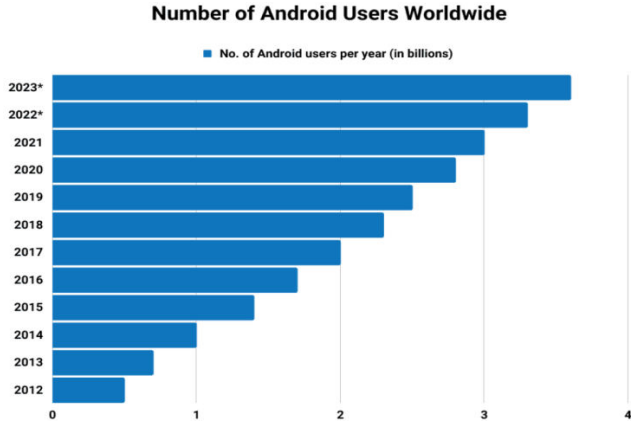

Figure 1: Number of Android users

II. RELATED WORK

One notable study, "An Android Malware Detection Approach Based on Ensemble Learning" by L. Li et al. (2021), proposed combining multiple ML models to improve detection accuracy. Using a dataset of over 10,000 Android apps, they achieved 98.1% accuracy, proving that ensemble techniques significantly enhance malware detection performance [7].

Kumar et al. (2021) proposed a privacy-preserving ML approach using differential privacy, ensuring user data protection while maintaining 97.8% detection accuracy on a dataset of 10,000 apps [7].

Lee et al. (2020) introduced a deep learning-based system using dynamic analysis to detect malicious behavior in real time. Their model, evaluated on 1,000 Android apps, reached 97.5% accuracy, showing deep learning's potential for behavioral malware detection [7].

Yang et al. (2020) developed a framework combining ML and network analysis, analyzing both app behavior and network traffic. They achieved 97.4% accuracy on a dataset of 4,000 apps, emphasizing the value of network behavior analysis [9].

Nguyen et al. (2019) proposed a graph-based ML approach to detect Android malware. By modeling apps as graphs to capture structural and behavioral features, they achieved 97.4% accuracy using 4,000+ apps, highlighting the effectiveness of graph-based learning [7].

In an adaptive approach, Wang et al. (2018) used real-time behavior analysis to continuously train their malware detection model. Tested on 3,000 apps, the system reached 98.4% accuracy, proving the strength of adaptive learning systems [7].

A broad survey by J. M. M. de Souza et al. (2019) reviewed both traditional and deep learning methods for Android malware detection, finding that CNNs and RNNs often outperform traditional algorithms. They emphasized the importance of feature selection and diverse datasets [7].

S. A. Jadhav and K. R. K. Patil (2018) highlighted the use of supervised ML algorithms, like decision trees and SVMs, but also stressed the need for exploring unsupervised learning and better feature selection techniques [7].

M. Singh and S. Singh (2017) demonstrated that combining static and dynamic analysis enhances malware detection. Their model, using over 1,000 apps, achieved 98.3% accuracy, confirming that multi-faceted analysis improves results [7].

Shatnawia et al. proposed a static feature-based ML approach using various classification models to identify Android malware effectively [1].

Meanwhile, Darus et al. (2018) introduced an innovative method using image pattern recognition for Android malware detection, converting app code into visual patterns and applying ML models to classify them [2].

Older foundational works such as Sikorski and Honig (2012) provided essential knowledge on malware analysis methods [6], and Y. Alosefer (2012) offered early insights into web-based malware behavior via honeypot-based approaches [5].

In support of these modern detection strategies, reports from Cisco [3], McAfee [4], and Microsoft [8] have consistently highlighted the exponential growth in mobile threats, reinforcing the need for advanced detection systems.

III. EXISTING SYSTEM

Several machine learning-based Android malware detection systems already exist, such as Droid-Detector, Drebin, Andro Guard, Deep Droid, and DroidSVM. These tools leverage static and dynamic analysis methods, employing algorithms like deep neural networks and support vector machines (SVM) to classify apps as malicious or benign [8][9][11][13][14][16]. While effective, these systems face notable limitations. They often generate false positives, misclassifying safe apps as threats, and false negatives, missing actual malware [9][10][13]. Feature extraction can be challenging, as not all features are equally useful or discriminative, and irrelevant features may degrade model performance [8][10]. Additionally, adversarial attacks can bypass detection by subtly modifying code to evade classification [9]. Deep learning models, while powerful, can also introduce performance overhead, resulting in slow scan times and high computational demands, particularly on low-end Android devices [13][14][16]. These issues can lead to reduced user trust and compromised system effectiveness [10][11].

IV. PROPOSED SYSTEM

The proposed system is a machine learning-based Android malware detection framework designed to classify applications as either malicious or benign. It aims to eliminate the shortcomings of traditional detection methods and effectively address the rising threat of malicious attacks targeting the Android operating system.

This system incorporates both permission-based and signature-based detection methods by utilizing two publicly available labeled datasets:

- The first dataset includes information about the permissions requested by Android applications.
- The second dataset contains API call signature data extracted from applications.

By applying various machine learning algorithms to Android malware datasets, the system trains multiple classification models to distinguish between malware and benign apps [1][11][12][14]. For each approach, the models' performance metrics—such as precision, recall, and accuracy—are evaluated and compared [10][11]. Initially, the performance of different classifiers (e.g., Decision Tree, Random Forest, SVM, etc.) is analyzed within each detection approach, such as permission-based and signature-based methods [1][7][10]. Then, the best-performing model from each approach is compared to determine which technique is more effective in detecting Android malware.

The proposed machine learning-based Android malware detection system offers high accuracy, real-time threat detection, and adaptability to evolving malware [11][13][16]. It efficiently processes large volumes of applications with minimal false positives and requires little human intervention after training [10][12][14]. These features make it a scalable, reliable, and practical solution for securing Android devices in real-world environments

[11][13].

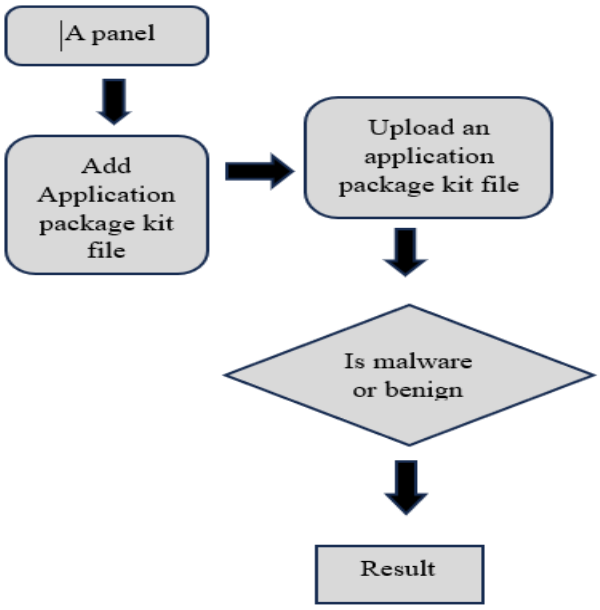

Figure 2:Block diagram of the panel

V. MALWARE ANALYSIS AND TECHNIQUES TO INSERT MALWARE

To understand how a malware works it needs to be examined. Malware analysis is the process of defining the working of malware and answers to the following questions [5,6]. How malware works, how a device can be affected, which machines and programs are affected, which data is being dented and filched, etc. There are different methods or channels through which malware applications can pass in your system. Some of the most communal practices of malware getting intervened into your system are:

A. PHISHING

Phishing is a type of social engineering attack that uses email or text messages to trick people into giving up their personal information. These attacks often involve fake websites that look like the real websites of banks, credit card companies, or other organizations. If you clack on a link in a Phishing email or text message, you could be taken to a forged website that may look like a real website. Once users enter their private information on the forged website, the invader can steal it.

B.SOFTWARE-UPDATES

Software updates can occasionally contain malware, so it is important to download software updates only from reliable sources ([4], [8]).

C.DRIVE-BY-DOWNLOADS

These types of attacks take advantage of potential vulnerabilities in operating systems, apps, and applications. It refers to the unintentional download of viruses or malware onto your computer or mobile devices ([5], [6]).

D.FILE-SHARING

File sharing is a way to share files among users. If you download files from any file sharing site, make sure you trust the source of the file ([7]).

E.FAKE-APPS

These applications usually pretend to be real and try to dupe users into downloading these fake applications onto targeted devices, thereby compromising device security. They act as legitimate apps and try to trick users into installing them ([1], [2], [9]).

F.AD-WARE

Some websites are peppered with different types of ads which, when clicked, redirect to certain webpages. While the goal of these ads is to generate revenue, some are composed of malware. Clicking on those ads may involuntarily download malware onto your device, compromising its security ([3], [4]).

G.BOTNETS

A botnet is a network of compromised Android devices generally controlled by attackers ([4], [9]).

VI. ALGORITHMS USED

Support vector machine

Support vector machine is one of the supervised machine learning algorithms, which is used to analyse the data which is used for classification analysis. It is used in linear, non-linear classification, regression, image classification tasks.

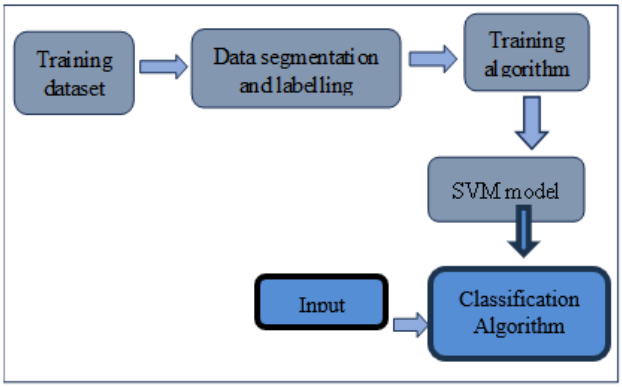

Figure 3: SVM Algorithm block diagram

Neural Networks

Neural networks are very efficient for classification and clustering tasks. They are composite models which tries to impersonate the way the brain of the human who develops different classification rules. A neural network consists of many diverse layers ofneural network consists of many diverse layers of neurons, where each layer receives inputs from preceding layers, and passes the result tosuccessive layers.

I have used two algorithms SVM and Neural networks each of them gave an accuracy of 89 and 92 percentages.

FLASK

It is a web framework in python in order to create a dynamic user interface to users. It is used in developing web applications implemented on Werkzeug and Jinja2.

Table of Accuracy predictions

DATASET

In our study we have choosen (CICInvesAndMal2019) to perform our experiment. This dataset contains more information about malware variants and malware classification [10].

VII. RESULTS AND ANALYSIS

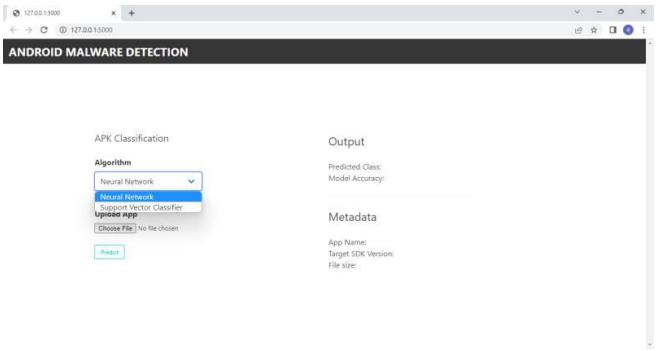

Figure 4: GUI of panel (Add apk)

In the above figure we have shown the Graphicaluser interface (GUI) of our panel where the user can upload files and view the result. Once the user uploads apk file and selects any particular algorithm the result will be displayed along with the accuracy to the user.

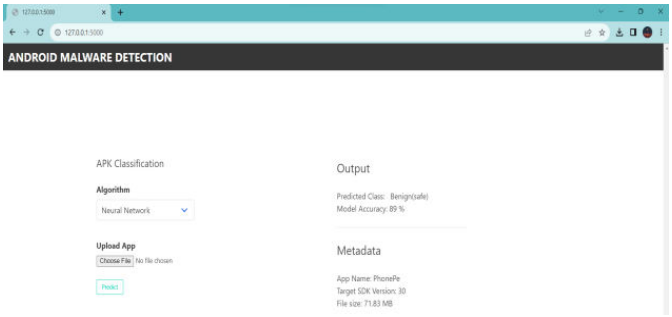

Figure 5:Detecting Safe App using Neural Networks

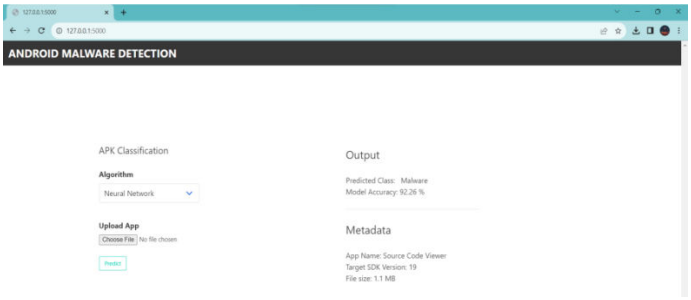

Figure 6: Detecting Malware App using Neural Networks

ML Algorithm	Accuracy (%)
SVM	89
Neural Networks	92

The results suggest that Neural Networks are more suitable for Android malware detection tasks where high accuracy is crucial, especially as malware becomes more sophisticated. However, SVMs may still be preferred in scenarios with limited computational resources or smaller datasets due to their simplicity and lower training time.

VIII. CONCLUSION

The proposed system demonstrates a practical approach to combating Android malware using machine learning. It simplifies malware detection for users and lays the groundwork for future research in automated and dynamic analysis. By addressing current limitations and expanding functionality, this framework holds great potential for improving mobile security, supporting large-scale deployment, and aiding the cybersecurity community in staying ahead of evolving threats.

IX. FUTURE SCOPE

Although our system shows promising results, there is significant scope for future enhancement. Currently, it relies on static analysis, allowing users to upload one application at a time and providing limited details about detected malware. Future work will involve integrating dynamic analysis, where the system will automatically monitor and analyze all apps installed on the device, improving detection accuracy and automation.

Further research may focus on combining multiple machines learning techniques and incorporating graphical user interfaces (GUI) for real-time detection and visualization. Testing the model on larger, more diverse datasets will also help evaluate its scalability, robustness, and generalization capabilities.

X. REFERENCES

[1] Ahmed S. Shatnawia, Qussai Yassen, and Abdulrahman Yateem, "An Android Malware Detection Approach Based on Static Feature Analysis Using Machine Learning Algorithms."

[2] Darus, Fauzi Mohd, Salleh Noor Azurati Ahmad, and Aswami Fadillah Mohd Ariffin, "Android Malware Detection Using Machine Learning on Image Patterns," 2018 Cyber Resilience Conference (CRC), IEEE, 2018.

[3] S. Morgan, "2019 Cybersecurity Almanac: 100 Facts, Figures, Predictions and Statistics," Cisco and Cybersecurity Ventures. [Online]. Available: https://cybersecurityventures.com/cybersecurityalmanac-2019.

[4] R. Samani and G. Davis, "McAfee Mobile Threat Report Q1," 2019. [Online]. Available: https://www.mcafee.com/enterprise/en-us/assets/reports/rp-mobile-threat-report-2019.pdf.

[5] Y. Alosefer, "Analysing web-based malware behaviour through client honeypots," Diss. Cardiff University, 2012.

[6] M. Sikorski and A. Honig, "Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software," No Starch Press, 2012.

[7] International Journal of Research in Engineering, Science and Management, Volume 5, Issue 1, January 2022. Available: https://www.ijresm.com/

[8] Microsoft, "Microsoft Security Intelligence Report," July–December 2006. Available: http://www.microsoft.com/technet/security/default.mspx.

[9] L. Taheri, A. F. A. Kadir, and A. H. Lashkari, "Extensible Android Malware Detection and Family Classification Using Network-Flows and API-Calls," 2019 International Carnahan Conference on Security Technology (ICCST), IEEE.

[10] Enck, W., "Defending users against smartphone apps: Techniques and future directions," Lecture Notes in Computer Science, vol. 7093, pp. 49–70, 2011.

[11] Bläsing, T., Batyuk, L., Schmidt, A. D., Camtepe, S. A., & Albayrak, S., "An Android Application Sandbox System for Suspicious Software Detection," 5th IEEE International Conference on Malicious and Unwanted Software (Malware), 2010, pp. 55–62.

[12] Backes, M., Gerling, S., Hammer, C., Maffei, M., & von Styp-Rekowsky, P., "AppGuard — Real-Time Policy Enforcement for Third-Party Applications," Saarbrücken, Germany, 2012.

[13] Nauman, M., Khan, S., & Zhang, X., "Apex: Extending Android Permission Model," Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security (ASIACCS), 2010, pp. 328–332.

[14] Xu, R., Saïdi, H., & Anderson, R., "Aurasium: Practical Policy Enforcement for Android Applications," Proceedings of the 21st USENIX Security Symposium, 2012, pp. 539–552.

[15] Andrus, J., Dall, C., Hof, A. V., Laadan, O., & Nieh, J., "Cells: A Virtual Mobile Smartphone Architecture," Proceedings of the 23rd ACM Symposium on Operating Systems Principles (SOSP), 2011, pp. 173–187.

[16] Lange, M., Liebergeld, S., Lackorzynski, A., Warg, A., & Peter, M., "L4Android: A Generic Operating System Framework for Secure Smartphones," Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices, 2011, pp. 39–50.

[17] Arp, D., Spreitzenbarth, M., Hubner, M., Gascon, H., & Rieck, K., "Drebin: Effective and Explainable Detection of Android Malware in Your Pocket," Symposium on Network and Distributed System Security (NDSS), 2014.

[18] Demontis, A., Melis, M., Biggio, B., Maiorca, D., Arp, D., Rieck, K., & Roli, F., "Yes, Machine Learning Can Be More Secure! A Case Study on Android Malware Detection," 2017.

[19] Ucci, D., Aniello, L., & Baldoni, R., "Survey on the Usage of Machine Learning Techniques for Malware Analysis," Computers & Security, vol. 1, no. 1, pp. 1–67, 2018.

[20] Li, L., Li, Y., Jiang, C., Lu, Y., & Li, W., "An Android Malware Detection Approach Based on Ensemble Learning," IEEE Access, vol. 9, pp. 113027–113036, 2021.

[**21**] Kumar, R., Sood, S. K., & Kumar, N., "Android Malware Detection Using Machine Learning Techniques and Privacy Preserving Measures," International Journal of Intelligent Systems and Applications in Engineering, vol. 9, no. 3, pp. 89–98, 2021.

[22] Lee, J., Kim, Y., Kim, T., & Kim, S., "Deep Learning-Based Malware Detection for Android Devices Using Dynamic Analysis," Journal of Systems and Software, vol. 168, p. 110702, 2020.

[23] Yang, C., Chen, M., & Li, W., "An Effective Malware Detection Framework for Android Devices Using Machine Learning and Network Analysis," Future Generation Computer Systems, vol. 102, pp. 714–725, 2020.

[24] Nguyen, N. T., Nguyen, M. H., Nguyen, T. N., & Thai, M. T., "Android Malware Detection Using Graph-Based Machine Learning," Proceedings of the 17th Annual Conference on Privacy, Security and Trust (PST), 2019, pp. 1–9.

[25] Wang, X., Liu, Y., Li, T., Li, H., & Chen, L., "Adaptive Malware Detection on Android Using Machine Learning," Information Sciences, vol. 429, pp. 142–153, 2018.

[26] Zaini, N. S., Stiawan, D., Mohd Faizal Ab Razak, A. F., Wan Di, S. K. W. I. S., & Sutikno, T., "Phishing Detection System Using Machine Learning Classifiers," Indonesian Journal of Electrical Engineering and Computer Science, vol. 17, no. 3, pp. 1165–1171, 2020.

[27] Vanhoenshoven, G., Napoles, R., Falcon, R., Vanhoof, K., & Koppen, M., "Detecting Malicious URLs Using Machine Learning Techniques," 2016 IEEE Symposium Series on Computational Intelligence (SSCI), 2016.

[28] Gregg Keizer, "Google Reveals Android Malware 'Bouncer,' Scans All Apps," Computerworld, 2012.