International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 04 | April - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Intelligence Technique for Fixing Vulnerabilities in Mobile Applications

[1]Mr.GANESH, [2] Mrs. MAGNA YADLAPALLI,
[1]Mr.Ganesh, M.sc CFIS, Department of Computer Science Engineering,
ganesh.sp711@gmail.com, 8939005173, Dr. MGR UNIVERSITY, Chennai, India
Mrs. Magna Yadlapalli, Assistant -Professor, Centre of Excellence in Digital Forensics, Chennai, India

---

ABSTRACT - The increasing reliance on mobile applications for both personal and business purposes have led to a rise in security vulnerabilities, posing significant risks to users and organizations alike. As mobile applications evolve in complexity, so do the methods employed by attackers to exploit weaknesses. Traditional security practices often fall short in addressing the unique challenges of mobile platforms. This paper explores advanced intelligence techniques for identifying, analyzing, and mitigating vulnerabilities in mobile applications. Using different types of algorithms such as Gaussian, K-neighbors, SVM, Naïve bayes & Random Forest Algorithm. By leveraging machine learning, artificial intelligence (AI), and behavioral analytics, this approach enables the proactive detection of security flaws and the implementation of adaptive defenses. Key techniques include static and dynamic analysis, anomaly detection, and automated penetration testing, which together enhance the ability to identify critical vulnerabilities early in the development lifecycle. The integration of AI-powered security tools not only streamlines the vulnerability assessment process but also empowers developers to prioritize remediation efforts based on the severity of the threat landscape. This paper aims to highlight the effectiveness of intelligence-driven security solutions in safeguarding mobile applications against emerging cyber threats and ensuring robust, secure user experiences.

KEYWORDS: Data collection, Vulnerability testing, Gaussian, K-neighbors, SVM, Random Forest, Naïve bayes algorithm.

I. INTRODUCTION:

Cyber security has become a primary area of immediate concern to computer scientists and network engineers that satisfactory solutions to many issues are in order. As a result of the rapid evolutions in technology developments and their inherent integrations in all aspects of our lifestyles, a variety of malware applications and their intended targets have become well studied and identified. Among the malware variety that has received attention lately is Android malware which is found to occupy considerable attention in the web world. Android is one of the most common operating systems that dominate the operating system market. In 2020, the Android system accounted for 85% of the total number of smartphones that harness the Android system as the operating system of choice.[1]

Previously the malware detection for mobile application is done through the machine learning algorithm using SVM and linear regression. The prediction is not much accurate and the dataset is not high. So, the prediction is not useful for the mobile application process.[2]

The paper scope for using intelligence techniques to fix vulnerabilities in mobile applications involves outlining key objectives, deliverables, methodologies, timelines, and resources. The scope defines what the project will achieve and how the AI techniques will be integrated into the mobile app development lifecycle to enhance security. In this paper, I'm using Gaussian, K-neighbours, SVM, Random Forest, Naïve bayes, all these algorithms are used to find out the vulnerabilities.[3]

II. LITERATURE REVIEW:

Niall McLaughlin, Jesus Martinez del Rincon, BoojoongKang, Suleiman Yerima, Paul Miller et al., [4] had proposed novel android malware detection system that uses a deep convolutional neural network (CNN). Malware classification is performed based on static analysis of the raw opcode sequence from a disassembled program. Features indicative of malware are automatically learned by the network from the raw opcode sequence thus removing the need for hand-engineered malware features.

Justin Sahs, Latifur Khan et al., [5] had proposed the recent emergence of mobile platforms capable of executing increasingly complex software and the rising ubiquity of using mobile platforms in sensitive

applications such as banking, there is a rising danger associated with malware targeted at mobile devices.

Kaijun Liu, Shengwei Xu, Guoai Xu, Haifeng Li et al., [6] had proposed the android applications are developing rapidly across the mobile ecosystem, but Android Malware is also emerging in an endless stream. Many researchers have studied the problem of Android malware detection and have put forward theories and methods from different Perspectives. Existing research suggests that machine learning is an effective and promising way to detect Android malware

Dong-Jie Wu, Ching-Hao Mao, Te-En Wei, Hahn-Ming Lee, Kuo-Ping Wu et al., [7] had proposed Android Malware Detection through Manifest and API Calls Tracing:Recently, the threat of Android malware is spreading rapidly, especially those repackaged Android malware. Although understanding Android malware using dynamic analysis can provide a comprehensive view, it is still subjected to high cost in environment deployment and manual efforts in investigation.

Zarni Aung, Win Zaw et al., [8] had proposed Mobile devices have become popular in our lives since they offer almost the same functionality as personal computers. Among them, Android-based mobile devices had appeared lately and, they were now an ideal target for attackers. Android-based smartphone users can get free applications from Android Application Market.. The proposed framework intends to develop a machine learning-based malware detection system on Android to detect malware applications and to enhance security and privacy of smartphone users.

Saha et al. (2021)., [9] had proposed and it developed machine learning models that predict which components of a mobile app are most likely to contain vulnerabilities. By analysing code complexity, historical vulnerabilities, and developer practices, these models help prioritize which areas of the app need to be reviewed more carefully, allowing developers to focus on the most critical parts of the app.

Kwon et al. (2022).,[10] had proposed AI-based anomaly detection tools, monitor the real-time behavior of mobile apps to detect abnormal activities. These tools identify potential threats such as unauthorized access, privilege escalation, or data exfiltration based on deviations from normal usage patterns. By employing AI, these tools can learn from past incidents and continuously improve their detection capabilities.

III. PROPOSED METHODOLOGY

This proposed model is to implement the android mobile software oriented process. In this project we implement the machine learning algorithms to predict the malware detection. The algorithms used in this software are K-Neighbours Classifier, Random forest and Gaussian Naive bayes. Using this algorithm we are finding the classification report, confusion matrix, accuracy score and kappa score.

Android Dataset:

tcp_packets	15038
dist_port_tcp	3514
external_ips	1434
vulume_bytes	2061210
udp_packets	38
source_app_packets	21720
remote_app_packets	18841
source_app_bytes	8615120
remote_app_bytes	2456160
source_app_packets.1	21720
dns_query_times	5095

Using these data collections & dataset to be analysed and gives the benigns in the report.

SYSTEM ARCHITECTURE:

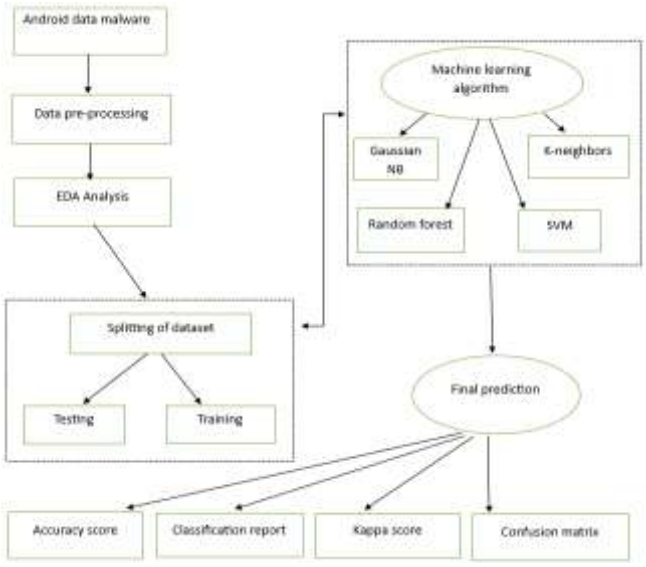

Fig 3.1 Architecture Diagram

Data Pre-processing:

In the pre-processing , by using the data.isna().sum() method to process and analysing the dataset.

source_app_packets 0
remote_app_packets 0
source_app_bytes 0
remote_app_bytes 0
duracion 7845
avg_local_pkt_rate 7845
avg_remote_pkt_rate 7845

dtype: int64

Splitting of Dataset:

Dataset can be splitted to analyse the dataset easily and findout the malicious & benigns.
It splitted into two sets Testing & Training.

Testing:

Comparatively testing the train dataset and analyse the data i.e, df = pd.read_csv("train.csv", sep=";").

Training:

Simultaneously training the android dataset i.e, data = pd.read_csv("android_traffic.csv", sep=";").

Final Prediction:

It consists of Accuracy score,confusion matrix,kappa score,Classification report.

Accuracy score:

Based on all algorithms, Random forest algorithm provides the best accuracy score,based on the score the malware should be rectified in the application.

Kappa score:

$$\kappa = \frac{P_o - P_e}{1 - P_e}$$

Where:

- Po is the observed agreement (the proportion of times the raters agree).
- Pe is the expected agreement by chance, which is based on the distribution of ratings across categories.

Classification report:

Classification report supports and classifies the table,scores,precision,recalls etc..

Confusion matrix:

A confusion matrix is a table used to evaluate the performance of a classification model, especially in terms of how well the model makes predictions for different classes.

Name	Precision	Recall	F1 score	Support	Accuracy
Gaussian NB	0.91	0.76	0.83	0.41	0.84
K-neighbours	0.85	0.95	0.96	41	0.89
Random forest	0.93	0.94	0.93	1190	0.92
SVM	0.81	0.12	0.20	1190	0.45

IV. FINDINGS:

Hence the Random forest algorithm gives the best accuracy score i.e,0.92.

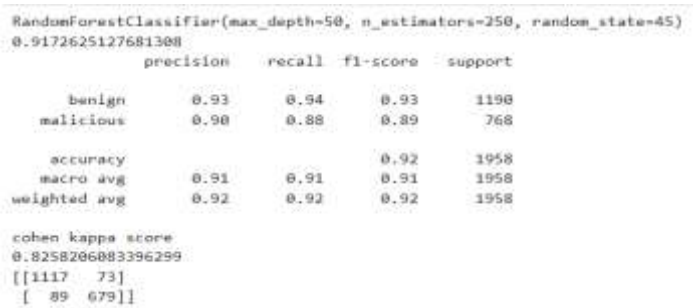

Fig 4.5 Random forest Classifier's Accuracy Score

Random Forest algorithm provides the best accuracy score from the analysis and the good benigns & least malicious scores

The accuracy score values are mentioned in above fig 4.5.

V. CONCLUSION: We provide the details of the steps that were taken to investigate the results that were obtained. We, also, analyze the performance of the various classifiers used in the study, showing the results for four different machine learning algorithms (Random forest, SVM, Gaussian naïve bayes, and K Neighbours Classifier) used in detecting malware with the machine learning algorithms. While sorting the vulnerabilities in the module we will add the AI enhancement for screen description to explain what we are enhancing and rebuilding, it will connect the user to understands what's happening in the future enhancements.

REFERENCES:

[1] P. Faruki, V. Ganmoor, L. Vijay, M. Gaur, and M. Conti, "Android Security: A Survey of Issues, Malware Penetration, and Defenses," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 998–1022, Secondquarter 2015. doi: 10.1109/COMST.2014.2386139.ACM Digital Library+2ACM Digital Library+2ACM Digital Library+2

[2] Teufl P, Ferk M, Fitzek A, Hein D, Kraxberger S, Orthacker C (2013) Malware detection by applying knowledge discovery processes to application metadata on the Android Market (Google Play). In: Security and communication networks. doi:10.1002/sec.675 [Online]. http://dx.doi.org/10.1002/sec.675. Accessed 1st April 2014

. [3] G. Suarez-Tangil and G. Stringhini, "Eight years of rider measurement in the android malware ecosystem: evolution and lessons learned," 2018, https://arxiv.org/abs/1801.08115.

[4] Shu LDong S(2025)Enhanced unknown Android Malware Detection using LG-PN: A local–global fusion approach in prototypical networksJournal of Information Security and Applications10.1016/j.jisa.2025.104062**91**(104062)Online publication date: Jun-2025 https://doi.org/10.1016/j.jisa.2025.104062

[5] W. Zhou, Y. Zhou, X. Jiang and P. Ning, "Detecting repackaged smartphone applications in third-party android marketplaces", *Proc. 2nd ACM CODASPY*, pp. 317-326, 2012. https://dl.acm.org/doi/abs/10.1145/2133601.2133640

[6] A. Reijonen, *The Evolution of Mobile Malware*, Bachelor's thesis, Jyväskylä University of Applied Sciences, 2024. Available: https://www.theseus.fi/handle/10024/851969

[7] R. Roy, S. Kumar, A. Tripathi, M. Das, and P. D. Dwivedi: Authors' initials and last names. *Food and Chemical Toxicology*: Name of the journal in italics.vol. 100: Volume number. pp. 1-8: Page range 2017: Year of publication. https://www.sciencedirect.com/science/article/abs/pii/S0016740481630164X

[8] L. Zhang, S. Zhang, F. Dong, W. Cai, J. Shan, X. Zhang, and S. Man: Authors' initials and last names. Food Research International: Name of the journal in italics. vol. 44, no. 5: Volume and issue numbers. pp. 1234–1241: Page range. 2011: Year of publication. doi: 10.1016/j.foodres.2011.01.002 https://www.sciencedirect.com/science/article/abs/pii/S1742287611000879.

[9] R. K. Gupta, S. S. Kanhere, and A. S. Bedi, published in *Sensors*, Volume 22, Issue 7, Article 2551 *Sensors* **2022**, *22*(7), 2551; https://doi.org/10.3390/s22072551https://www.mdpi.com/1424-8220/22/7/2551

[10] HyunJin Kim and Taeshik Shon, published in *The Journal of Supercomputing*, Volume 78, Pages 13554–13563, in 2022, https://doi.org/10.1016/j.cose.2016.11.011https://link.springer.com/article/10.1007/s11227-022-04408-4

[11] O. Olukoya, L. Mackenzie, and I. Omoronyia, "Towards using unstructured user input request for malware detection," Computers & Security, vol. 93, article 101783, 2020.

[12] P. Ravi Kiran Varma, K. P. Raj, and K. V. S. Raju, "Android mobile security by detecting and classification of malware based on permissions using machine learning algorithms," in 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), pp. 294–299, Palladam, India, 2017. 42

[13] J. Song, C. Han, K. Wang, J. Zhao, R. Ranjan, and L. Wang, "An integrated static detection and analysis framework for android," Pervasive and Mobile Computing, vol. 32, pp. 15– 25, 2016.

[14] S. Hahn, M. Protsenko, and T. Müller, "Comparative evaluation of machine learningbased malware detection on android," in Sicherheit 2016-Sicherheit, Schutz und Zuverlässigkeit, Gesellschaft für Informatik e.V, 2016.

[15] Q. Fang, X. Yang, and C. Ji, "A hybrid detection method for android malware," in 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), pp. 2127–2132, Chengdu, China, 2019.