International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 9, Special Issue INSPIRE'25, pp 24-32, April-2025
https://doi.org/10.47001/IRJIET/2025.INSPIRE04
International Conference on Sustainable Practices and Innovations in Research and Engineering (INSPIRE'25)

Android Malware Detection Using Machine Learning Techniques

[1]Shaik Salam, [2]Kalluru Pavankumar Reddy

[1]Associate Professor, Department of CSE, Mohan Babu University, Tirupathi, India
[2]MCA, Department of Computer Application, Mohan Babu University, Tirupathi, India. E-mail: pavanreddy141999@gmail.com

Abstract **- The rapid increase of Android application creation results in essential malware risks that need suitable detection systems to counteract these threats effectively. A machine learning-based Android Malware Detection System stands as the main component of this project development for classifying applications into benign versus malicious types. An evaluation of malware detection takes place through XGBoost (XGB) and Random Forest (RF) and Decision Tree (DT) algorithms by focusing on network patterns as well as API calls and permissions features. The training together with evaluation process for this system relies on a set of classified Android applications. The model achieves higher performance levels together with fewer selected features which produces simplified operational processes. The selection of the best real-time malware detector depends on F1-score calculations after performing accuracy tests and precision checks with recall validations. This study conducts an orderly examination of basic learning principles and ensemble learning principles to determine their major functions and constraints. Better Android security systems emerge from research results that lead to better malware detection algorithms that both achieve high accuracy and reduce false results. Security experts and developers obtain benefits from research implementation by building safer mobile applications. The research incorporates a machine learning system to improve Android security through specified tactics for detecting new malware threats.**

Keywords: Android malware detection, machine learning, Decision Tree, Random Forest, XGBoost, cybersecurity, mobile security, malware classification, feature selection, ensemble learning, Android security.

I. INTRODUCTION

The quick advancement of today's mobile technology results from Android's popularity in smartphone devices which have become essential tools for people's daily lives. Android dominates as the leading mobile system platform worldwide because the operating system continues to exist in billions of devices until 2025 [1]. Complex malware access points are through Android devices due to their universal use making attackers seek monetary gains by stealing information through unauthorized system penetration. Current Android malware has evolved to threaten user privacy along with security because signature-based detection fails to keep up [2]. Signature-based detection tools that provide malware defense cannot protect against modern malicious programs because both anonymous malware and automatically adjusted malware versions remain invisible to these tools. Researcher development of advanced detection systems now remains essential because existing detection methods show inadequacy in monitoring illegal software activities. Machine learning performs effectively as a malware detection method since it helps analyze large data volumes while targeting malicious operation signatures per [4]. ML-based techniques use their capability to understand known malware information for detecting new threats which leads to more reliable security systems. The precise classification of malware depends on analyzing features of applications with permission data and API functions and network activity that ML models can interpret. Data collection entails gathering a comprehensive dataset of both benign and malicious applications. The data extraction process selects important features that enable observation of hazardous application differences from common system programs. During training of ML algorithms its features help recognize established patterns which correspond to malware specifications. System performance evaluation is achieved through malware detection accuracy testing using accuracy, precision, recall and F1-score metrics according to [5] for model assessment. The continuous updates required for malware models stem from the fact that modifications made to malware infrastructure lead to permanent changes. Checking features properly stands as a fundamental process since inappropriate features reduce the model's operational performance. A detection framework must establish an optimal equilibrium between successful result generation and efficient resource utilization within mobile environmental restrictions. Researchers improved the detection accuracy through studies on how dynamic analysis elements operate with system calls in runtime systems [1]. Research teams examined static analysis by studying application codes through metadata to identify dangerous

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 9, Special Issue INSPIRE'25, pp 24-32, April-2025
https://doi.org/10.47001/IRJIET/2025.INSPIRE04
International Conference on Sustainable Practices and Innovations in Research and Engineering (INSPIRE'25)

patterns that detected in these applications. According to experts static and dynamic analysis testing should be combined since this approach utilizes both methods to their fullest potential. Researchers indicate the detection systems employing ML need the widespread data found in Drebin dataset to test their effectiveness [3]. The detection system performance assessment relies on existing datasets to ensure precision during modeling development and performance testing phase. The emergence of new malware encourages scientists to maintain an ongoing program for method identification research. The threat identification process for unknown security threats works more effectively utilizing ML-based detection systems rather than signature-based detection methods. The protection of Android users who make up the wide population requires ongoing research to outpace sophisticated malware.

II. LITERATURE SURVEY

According to Ayat Droos et al. [6], Cosmos system employs machine learning algorithms to search for Android malware through assessments of APK instances in CICMalDroid2020 dataset. The Random Forest model provided the best results for classification assessment with 27 selected key features resulting in a 98.6% accuracy rate. The system achieves its maximum detection performance by implementing feature selection but Android malware continues to grow as a threat against devices therefore requiring active security development.

Naseef Chowdhury combined forces with colleagues to deeply inspect machine learning techniques used for Android malware identification in their research paper [7]. This research divides malware detection techniques into three distinct segments including supervised and unsupervised together with deep learning which successfully detect malware. Multiple detection systems undergo performance assessments through researcher evaluation of measurement criteria that determine their detection accuracy. The article explores active malware development challenges and presents research directions for developing better protection systems to enhance mobile device security.

Amarjyoti Pathak's research team established an Android malware detection system which utilized important feature scores for selecting attributes according to their work [8]. The research project first employs Gradient Boosting to select significant permissions then uses these features to reduce the vector dimensionality. The performance of training capabilities improves through reduced feature vectors without affecting accuracy results. Better detection of malware depends on feature selection optimization since it allows for faster execution matching real-world needs.

Android malware detection strategies with machine learning received scrutiny from Ebtesam J. Alqahtani et al. [9] to increase their accuracy performance levels. This research evaluates the ability of Support Vector Machines (SVM), Naïve Bayes (NB) and Deep Neural Networks (DNN) for detecting malicious software by using them as detection techniques. The research demonstrates machine learning provides Android devices with their needed defense against security threats since their systems urgently need improvement to protect against malware threats that increase due to an expanding user base.

The detection system proposed by F. A. Almarshad and colleagues combines Siamese Shot Learning with machine learning approaches according to their research paper [10]. Multiple malware samples can be categorized using Siamese neural networks which operate under the one-shot learning framework when receiving constrained input data. During the Drebin data testing phase the proposed model achieved an accuracy of 98.9% surpassing traditional detection methods. The examined studies show that deep learning solutions boost Android system security features effectively.

A systematic Android malware machine learning detection evaluation was presented in the article written by Janaka Senanayake and his research team [11]. The review of precisely chosen 106 studies examines their positives and negatives while identifying future development areas. The detection success of machine learning systems reaches its peak for Android devices because these phones represent the majority of mobile devices in the market. As a part of this paper researchers discuss diverse techniques to detect program vulnerabilities in source code along with emphasizing predeployment security assessments. Scientists will use results from this research to determine the path forward for Android malware monitoring technology development.

The combined paper of Ali Muzaffar et al. [12] delivers an extensive study of Android malware detection based on machine learning techniques with assessments of performance and detection obstacle identification. This technology creates security advantages by fixing the essential flaw which_signature-based systems experience when monitoring zero-day attacks. The paper evaluates supervised and unsupervised learning methods as well as deep learning and online learning approaches through static and dynamic and hybrid characteristics categorization. Machine learning demonstrates promising potential to secure the Android platform because of its growing threats and extensive platform usage.

In their research Halil Murat Ünver et al. [13] established new Android malware detection through machine learning

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 9, Special Issue INSPIRE'25, pp 24-32, April-2025
https://doi.org/10.47001/IRJIET/2025.INSPIRE04
International Conference on Sustainable Practices and Innovations in Research and Engineering (INSPIRE'25)

with image-based features. eskortech computing transformation generates properties from four specific local elements and three global elements that the model can process. Random Forest together with K-Nearest Neighbors, Decision Tree, Bagging, AdaBoost and Gradient Boost form a combination of machine learning classifiers that train their algorithms by using application file features. The proposed methodology achieved a testing duration of 0.018 seconds with 98.75% accuracy to classify samples which surpassed the existing performance standards in terms of speed and accuracy metrics.

A system for Android malware detection through machine learning was described by K. S. Ujjwal Reddy et al. in their work [14] while using application permissions and activities as analysis inputs. The Android application installation phase makes it possible for the Androguard tool to recognize specific features of installed apps. Without running any dynamic test the solution determines whether an application is safe or dangerous. The model underwent training by being provided with 1000 applications where 500 were malicious and 500 were benign. Random Forest demonstrated superior success rates totaling 95% while K-Nearest Neighbors stopped at 79%.

Hashida Haidros Rahima Manzil et al. [15] designed a modern Android malware detection system which detects adware in addition to riskware and banking malware and SMS malware variants. The Huffman encoding method and system call frequencies provide the base for generating malware dynamic pattern features in this model. The analysis of the suggested model generated 98.70% detection accuracy by utilizing Random Forest classifier to outperform existing research methods.

Nahier Aldhafferi et al. [16] presented Support Vector Regression (SVR) with dynamic feature analysis for Android malware detection that functioned as a novel solution. Researchers evaluated the effect of using SVR-enabled Radial Basis Function Kernel for malware detection by extracting features from dynamic Android applications. Experimental results for the model model produced accuracy of 95.74% and precision of 94.76% and recall of 98.06% which led to an F1-score of 96.38%. The adopted research method displayed better discrimination power than SVM and Random Forest and CNN.

Rishab Agrawal et al. studied Android malware detection through signature-based approach limitations to identify new threats according to their paper [17]. Research showed that machine learning algorithms combined with semantical analysis should serve as the analytical method for analyzing Android malware. After performing a scan which matches the detected application functions to known functions the method obtains permissions for the user. The system enables users to examine dangerous permissions in apps before adding their comments which strengthens the detection of new malware bets.

Through decompiled Android smali packages Burak Tahtaci et al. [18] developed a system for Android malware detection by extracting n-gram features. Research investigations concentrated on mobile application safety threats because these threats generate more frequent situations of data theft and identity fraud. The authors used various methods to build automatic malware detection models while eliminating manual work through their different feature selection approaches. Research has proven that machine learning makes malware inspection tools more effective according to this study.

Shaikha Al Ali et al. [19] created an Android malware detection system which incorporated KNN, DT, NB, SVM and ensemble classifiers XGBoost, LGBM, and CatBoost. Hyperparameters within the SVM system achieved 99.5% accuracy when radial basis function (RBF) operated as its kernel. Multiple research signals to the community that machine learning presents a successful method of improving mobile security because it detects sophisticated malware infiltrations.

A new Android malware detection system was created by Esraa Odat and her co-authors who analyzed static features within abnormal permission lists and API request signatures of malware during their development process [20]. The researchers evaluated Drebin and both Malgenome and MalDroid2020 malware datasets through FP-growth algorithm to identify particular co-occurring attributes. Random Forest achieved 98% maximum accuracy in evaluation when testing at the second permission combination level thus delivering superior results to existing methods.

III. PROPOSED SYSTEM

This system focuses on Android malware detection by having state-of-the-art machine learning techniques cooperate with dynamic feature analysis techniques. The system provides better detection than signature-based approaches because it reveals both known and unknown forms of malware. Real-time application tracking occurs through the system by extracting dynamic features that incorporate system calls and API calls along with permissions. A new method of boosting malware classification accuracy is achieved by the system through implementation of Huffman encoding for feature vector generation. The chosen classifiers encompass Decision Tree (DT) and Random Forest (RF) while XGBoost

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 9, Special Issue INSPIRE'25, pp 24-32, April-2025
https://doi.org/10.47001/IRJIET/2025.INSPIRE04
International Conference on Sustainable Practices and Innovations in Research and Engineering (INSPIRE'25)

(XGB) exists as well because they demonstrate great effectiveness for handling the non-linear relations that arise within the feature domain. The model evaluates its effectiveness using Drebin, Malgenome and MalDroid2020 database results and reaches maximum detection success rates of 98%. The system implements the hybrid strategy in combination with machine learning algorithm coexistence as a method to boost detection precision and lower false positive rates. The model achieves superior benign and malicious application separation by integrating static and dynamic features together. The research approach lays down fundamental principles for building malware detection frameworks for Android devices to develop scalable mobile security systems that achieve high accuracy rates in malware protection..

IV. DATASET

An effective malware detection system emerges from using both static and dynamic features obtained from Android applications. Three distinguished sources Drebin, Malgenome, and MalDroid2020 provide benign and malicious Android applications to construct the dataset from their databases. Multiple necessary features were integrated into the dataset to monitor application dynamic behavior by tracking permissions API calls along with system calls. These extended features in the dataset represent permission (API call) pattern combinations at different levels that enhance the detection capabilities for complex malware behaviors. Features are selected from these elements through the combination of Frequent Pattern Growth (FP-growth) algorithm and it's supporting techniques in the processing system. A bias-eliminating balance has been incorporated into the dataset to guarantee both unseen data generalization and improved reliability in malware detection.

V. METHODOLOGY

A systematic development of Android malware detection software is explained using Decision Tree (DT) with Random Forest (RF) and XGBoost (XGB) machine learning algorithms. The research purpose aims to categorize Android applications between benign and malicious types using static and dynamic feature examination of their data. The malware detection system development proceeds through multiple formal stages that support its assessment together with development process.

1. Data Collection and Preprocessing: serves as the first important element in this project framework. The study utilizes Drebin Malgenome and MalDroid2020 datasets for its analysis of benign and malicious Android applications that research communities consider validated. The datasets play an essential part in the system since they contain different malware categories encompassing adware combined with spyware and banking malware as well as riskware. The dataset samples include different features indicating both application permission requests and API calls as well as system calls and application preprocessing elements before data handling happens. Prior to data processing the approach performs duplicate data elimination while completing missing data points processing as well as converting categorical information to numeric values. After splitting the data the information rests in two distinct parts named training data and testing data. A different portion of data serves for testing after the model executes training tasks with its initial set to ensure both overfitting prevention and model generalization confirmation.

2. Malware detection development: needs feature extraction as its core process for machine learning model implementation. Both static features receive attention along with dynamic features during analysis in the current project. The static features of Android APK files appear prior to app execution without running the software. The framework contains three fundamental data elements which are app permissions requests and made API queries together with permission-API pattern co-occurrences data. Applications require Permissions to connect with operating system system components for exchanging data and they support contact reading functions and SMS sending capabilities as well as internet access. A formal malware detection algorithm can run through observing permissions which exceed required functions for normal applications. The asking of multiple permission requests by an app function as an obvious danger indicator. The detection system utilizes two types of features to track system call usage and examine API behavior as well as observing program behavior during execution. Static and dynamic features allow the model to build an extensive understanding of the way apps behave. The threat-oriented applications utilize multiple API transmission sequences accompanied by permission approval settings. A software program which asks for SMS and internet permissions contains suspicious purposes. The measurement tool creates a system of relationship identification through multiple levels beginning with second level then third level and continuing to fourth level and finishing with fifth level.

3. The system needs to choose appropriate groups: Extracted features when identifying suitable mobile malware classification parameters. The process which eliminates unrequired or matching attributes during analysis belongs to feature selection. Association rule mining through the Frequent Pattern Growth (FP-growth) algorithm identifies the most crucial features which exist together. FP-growth detects

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 9, Special Issue INSPIRE'25, pp 24-32, April-2025
https://doi.org/10.47001/IRJIET/2025.INSPIRE04
International Conference on Sustainable Practices and Innovations in Research and Engineering (INSPIRE'25)

common data patterns by identifying situations where features match each other beyond statistically natural levels of occurrence. PCA techniques reduce the number of variables in the dataset while maintaining vital details found in the chosen features. The features undergo normalization step after which they attain standardized range boundaries. The XGBoost algorithms together with Decision Trees need normalized features because both methods detect changes that occur from different feature scales. The endpoint performance and the models' convergence speed increase as a result of this process.

4. The evaluation of Android malware: uses three detection algorithms which consist of Decision Tree (DT) and Random Forest (RF) with XGBoost (XGB as part of training and evaluation. The selected algorithms manage both structured and unstructured information with great success during classification activities.

As a supervised learning method the Decision Tree algorithm uses a system for breaking down data entries to identify the most important attributes. Repeated recursive divisions on the data produce feature-driven decision points until the algorithm determines the class classification nodes as benign or malicious. The splitting process depends on a Gini impurity or information gain criterion to determine the capability of features to differentiate different classes.

The interpretability of Decision Trees remains exceptional because their basic design provides easy understanding to users. Decision Trees provide simple to understand visual displays which allow users to observe important features during classification procedure. The depth of tree structures leads to overfitting due to their architectural characteristics. A solution to this issue needs pruning techniques to simplify the structure of the tree.

Through ensemble learning Random Forest builds various decision trees so as to obtain better predictive accuracy while reducing overfitting. Random Forest operates by using a collection of Decision Trees which generate predictions from randomly sampled data parts. A collective prediction from the entire forest of trees is consolidated by majority vote before final classification. Random Forest provides exceptional handling for large datasets encompassing many dimensions while its aggregated forests minimize data overfitting. The aggregation of multiple trees in Random Forest surpasses Decision Tree performance because it controls prediction variance by combining diverse trees.

XGBoost: works as a gradient-boosted decision tree procedure which produces one sequence of decision trees to fix errors found in prior predictions. Operation speed and accuracy performance stand as the main benefits of XGBoost. XGBoost reaches minimum loss function performance by using gradient descent optimization which makes it effective at finding malware in unbalanced datasets. The advantages XGBoost brings to machine learning include value imputation together with regularization algorithms for overfitting restriction and parallelized training implementations. XGBoost obtains its exceptional power for pattern classification through the integration of several usable features.

5. Model Evaluation requires: The use of accuracy and precision as well as recall and F1-score together with Area Under Curve (AUC) for evaluating trained models. A model demonstrates its accuracy through score measurements but precision and recall information shows how well it identifies suspicious and non-threatening applications accurately. Model evaluators calculate the weighted average between precision and recall by performing harmonic mean calculations that produce the F1-score. Model performance regarding its ability to detect malicious versus benign applications is assessed by AUC-ROC curves as cross-validation creates stable models through selection of training subsets. The model splitting process distributes data into separate folds which enables different data subsets to train the model as it tests its capacity to detect unknown samples.

6. The XGB, DT and RF algorithms demonstrate: The best performance as the most desirable Android malware detection algorithm from model testing results. The decision-making process for selecting a final model relies on the achievement of optimal performance metrics by a particular model. During this phase the team conducts investigations about data classification methods of each algorithm and both the selected features used in decisions and systematic strengths together with weaknesses are assessed.

7. Real Android applications serve as test subjects in the laboratory to check how the selected model functions for unsuspected Android applications. The authentication process of every input application checks its status and generates a prediction confidence rating in addition to the benign or malicious determination outcome. Users can evaluate the hazard potential of an app by referring to the score provided by the system.

8. The model achieves performance enhancement through grid search and random search algorithms applied for hyperparameter optimization. These optimization techniques help identify superior combinations between selected parameters which consist of learning rate parameters and tree depth and tree count. By using this methodology users can create Android malware detection systems through machine learning methods with optimally functioning and accurate

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 9, Special Issue INSPIRE'25, pp 24-32, April-2025
https://doi.org/10.47001/IRJIET/2025.INSPIRE04
International Conference on Sustainable Practices and Innovations in Research and Engineering (INSPIRE'25)

performance capability. The system analyzes Android malware by utilizing Decision Tree and Random Forest with XGBoost for the evaluation of static and dynamic features along with co-existing features. Performance metrics allow researchers to identify the best model which becomes the candidate for implementation. The approach has developed a security system that effectively finds Android malware in order to enhance mobile device protection.

VI. RESULTS AND DISCUSSIONS

The Android malware detection framework uses Decision Tree (DT) Random Forest (RF) and XGBoost (XGB) as algorithms that produced excellent results for assessment scores as well as precision recall and F1-score metrics. The accuracy rates computed from detected algorithms showed that Decision Tree achieved 98% accuracy while Random Forest and XGBoost reached 98% and 99% respectively. All models demonstrated an outstanding capability to detect genuine Android apps among malicious ones according to the results. The high accuracy rates of XGBoost led it to become the best approach for distinguishing legitimate programs from dangerous malware programs between the two categories. The model performance evaluation depends on confusion matrix data including true positive (TP) and false positive (FP) counts together with true negative (TN) and false negative (FN) numbers. Decision Tree alongside Random Forest and XGBoost show excellent capabilities for precise identification of both benign and malicious applications according to the confusion matrices. A low total number of errors across the evaluation enables malicious applications to receive accurate detection while ensuring no incorrect malicious detection occurs for benign applications. The precision value of Decision Tree reached 95.8% and its recall performance surpassed 96.9% and resulted in an F1-score of 96.3%. Test results demonstrated how Random Forest produced 96.6% precision together with 97.0% F1-score along with 97.5% recall. XGBoost demonstrated peak efficiency through its performance metrics which comprised 96.6% precision and 98.5% recall and 97.5% F1-score. These performance metrics prove that XGBoost stands away as the most trustworthy malware detector because it has a minimal chance of incorrectly identifying applications. The exceptional performance metrics of Random Forest MODEL occurred because this algorithm demonstrated both high accuracy levels and recall performance qualities simultaneously. The prediction capacity of Decision Tree remained lower but it provided an appropriate solution approach for malware detection tasks.

The evaluation based on AUC-ROC curves indicates XGBoost models deliver the foremost detection capabilities. XGBoost achieved 0.99 as AUC value but Random Forest scored 0.98 and Decision Tree resulted in an AUC score of 0.97. The AUC indicators from numerous sources validate XGBoost as a superior detection system than Decision Tree and Random Forest because it demonstrates stronger abilities to differentiate malicious from legitimate software.

Android malware detection performance benefits most from using XGBoost which establishes its position as the optimal solution for this application. Decision Trees along with Random Forest algorithms stand as suitable alternative models for systems that seek different assessment guidelines like model transparency as well as execution speed efficiency.

Table 1: Model Comparison Table

Model	Accuracy	Precision	Recall	F1-Score	AUC
Decision Tree	98%	95.8%	96.9%	96.3%	0.97
Random Forest	98%	96.6%	97.5%	97.0%	0.98
XGBoost	99%	96.6%	98.5%	97.5%	0.99

The following confusion matrix images illustrate the performance of each model in more detail.

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 9, Special Issue INSPIRE'25, pp 24-32, April-2025
https://doi.org/10.47001/IRJIET/2025.INSPIRE04
International Conference on Sustainable Practices and Innovations in Research and Engineering (INSPIRE'25)

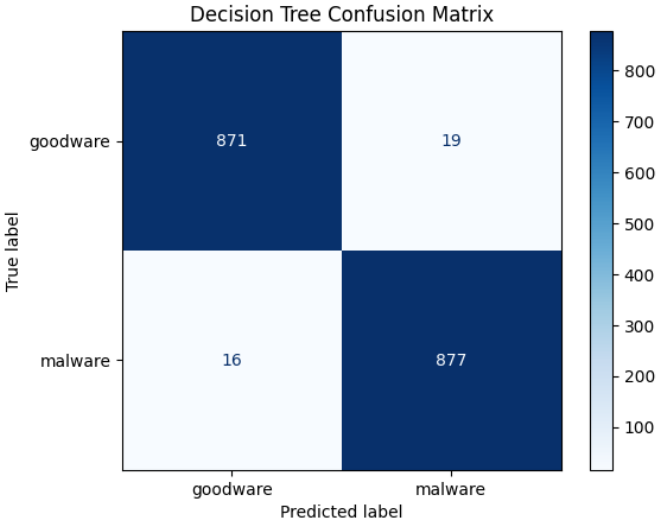

Fig. 1: Decision Tree Confusion Matrix

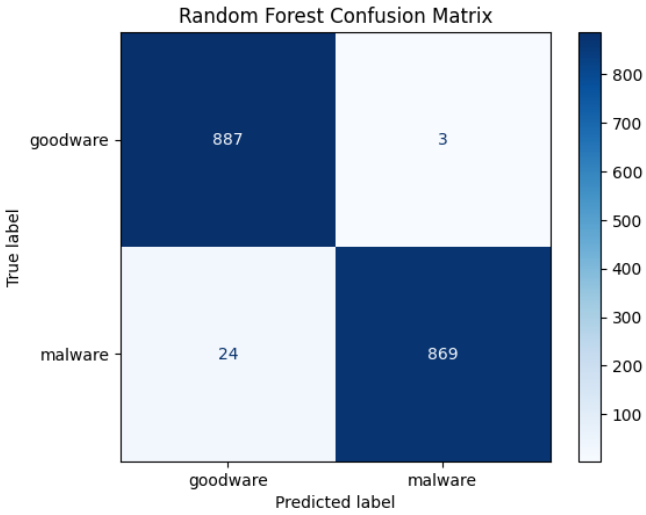

Fig. 2: Random Forest Confusion Matrix

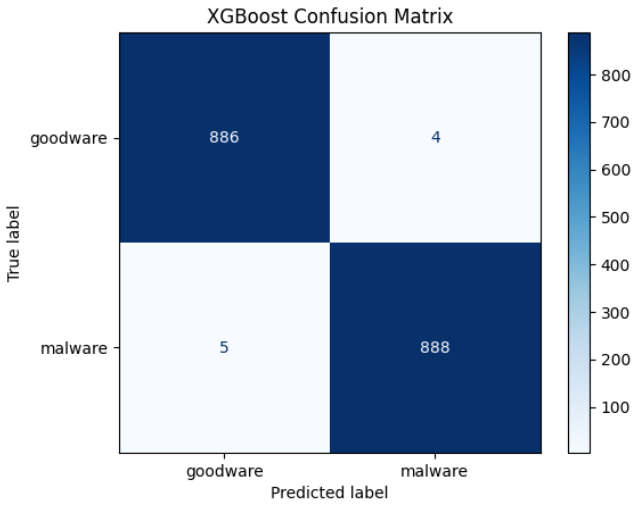

Fig. 3: XGBoost Confusion Matrix

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 9, Special Issue INSPIRE'25, pp 24-32, April-2025
https://doi.org/10.47001/IRJIET/2025.INSPIRE04
International Conference on Sustainable Practices and Innovations in Research and Engineering (INSPIRE'25)

Random Forest's multiple decision tree configuration delivers excellent malware detection capabilities which means it represents an optimal malware detection system option. The system's two important features ensure robust overfitting protection and quick handling of large data quantities leading to top performance. Even though Decision Tree shows lower malware detection achievement compared to other models it delivers dependable results in addition to great user accessibility and explanation features.

Security systems demonstrate the effectiveness of machine learning algorithms in detecting Android malware according to research data. All three models exhibit exceptional performance which shows they would be useful in actual Android malware protection systems.

VII. CONCLUSION

Testing proves that Decision Tree, Random Forest and XGBoost show excellent power when detecting Android malware. Machine learning algorithms show exceptional skill in separating beneficial applications from harmful ones through their exceptional accuracy measures as well as precision and recall and F1-score and AUC values.

The algorithmic performance of XGBoost reached 99% accuracy that outmatched both Decision Tree and Random Forest which registered 98% accuracy. XGBoost ensemble learning establishes a better generalization through multiple decision trees which yields improved classification performance because of its decision tree combination method.Random Forest's multiple decision tree configuration delivers excellent malware detection capabilities which means it represents an optimal malware detection system option. The system's two important features ensure robust overfitting protection and quick handling of large data quantities leading to top performance. Even though Decision Tree shows lower malware detection achievement compared to other models it delivers dependable results in addition to great user accessibility and explanation features.

Security systems demonstrate the effectiveness of machine learning algorithms in detecting Android malware according to research data. All three models exhibit exceptional performance which shows they would be useful in actual Android malware protection systems.

REFERENCES

[1] M. N.-U.-R. Chowdhury et al., "Android Malware Detection using Machine learning: A Review," arXiv preprint arXiv:2307.02412, 2023.

[2] A.Hefter, C. Sendner, and A. Dmitrienko, "Metadata-based Malware Detection on Android using Machine Learning," arXiv preprint arXiv:2307.08547, 2023.

[3] H. Papadopoulos et al., "Android Malware Detection with Unbiased Confidence Guarantees," arXiv preprint arXiv:2312.11559, 2023.

[4] J. Liu et al., "Unraveling the Key of Machine Learning Solutions for Android Malware Detection," arXiv preprint arXiv:2402.02953, 2024.

[5] A.Shabtai et al., "Andromaly: a behavioral malware detection framework for android devices," Journal of Intelligent Information Systems, vol. 38, no. 1, pp. 161-190, 2012.

[6] A.Droos, A. Al-Mahadeen, T. Al-Harasis, R. Al-Attar, and M. Ababneh, "Android Malware Detection Using Machine Learning," 2022 13th International Conference on Information and Communication Systems (ICICS), 21-23 June 2022, DOI: 10.1109/ICICS55353.2022.9811130.

[7] N. Chowdhury, A. Haque, H. Soliman, and M. S. Hossen, "Android Malware Detection using Machine Learning: A Review," TechRxiv, Apr. 2023, DOI: 10.36227/techrxiv.22580881.v1.

[8] A.Pathak, U. Barman, and T. S. Kumar, "Machine learning approach to detect android malware using feature-selection based on feature importance score," Journal of Engineering Research, Apr. 2024, DOI: 10.1016/j.jer.2024.04.008.

[9] E. J. Alqahtani, R. Zagrouba, and A. Almuhaideb, "A Survey on Android Malware Detection Techniques Using Machine Learning Algorithms," 2019 Sixth International Conference on Software Defined Systems (SDS), 10-13 June 2019, DOI: 10.1109/SDS.2019.8768729.

[10] F. A. Almarshad, M. Zakariah, and G. A. Gashgari, "Detection of Android Malware Using Machine Learning and Siamese Shot Learning Technique for Security," IEEE Access, vol. PP, no. 99, pp. 1-1, Jan. 2023, DOI: 10.1109/ACCESS.2023.3331739.

[11] J. Senanayake, H. Kalutarage, and M. O. Al-Kadri, "Android Mobile Malware Detection Using Machine Learning: A Systematic Review," Electronics, vol. 10, no. 13, p. 1606, 2021, DOI: 10.3390/electronics10131606.

[12] A.Muzaffar, H. R. Hassen, M. A. Lones, and H. Zantout, "An in-depth review of machine learning based Android malware detection," Computers & Security, vol. 121, p. 102833, Oct. 2022, DOI: 10.1016/j.cose.2022.102833.

[13] H. M. Ünver and K. Bakour, "Android malware detection based on image-based features and machine

learning techniques," SN Appl. Sci., vol. 2, p. 1299, 2020, DOI: 10.1007/s42452-020-3132-2.

[14] K. S. U. Reddy, S. S. C. Chakkaravarthy, M. Gopinath, and A. Mitra, "A Study on Android Malware Detection Using Machine Learning Algorithms," in ICISML 2022. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 470, Springer, Cham, 2022, DOI: 10.1007/978-3-031-35078-8_20.

[15] H. H. Rahima Manzil and S. M. Naik, "Android malware category detection using a novel feature vector-based machine learning model," Cybersecurity, vol. 6, no. 6, 2023, doi: 10.1186/s42400-023-00139-y.

[16] N. Aldhafferi, "Android Malware Detection Using Support Vector Regression for Dynamic Feature Analysis," Information, vol. 15, no. 10, 658, 2024, doi: 10.3390/info15100658.

[17] R. Agrawal, V. Shah, S. Chavan, G. Gourshete, and N. Shaikh, "Android Malware Detection Using Machine Learning," in 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), 24-25 Feb. 2020, doi: 10.1109/ic-

[18] B. Tahtaci and B. Canbay, "Android Malware Detection Using Machine Learning," in 2020 Innovations in Intelligent Systems and Applications Conference (ASYU), 15-17 Oct. 2020, doi: 10.1109/ASYU50717.2020.9259834.

[19] S. Al Ali, A. Suleiman, G. Hallal, S. Alseiari, Y. Ma, and S. Dhou, "Android Malware Detection Using Machine Learning," in 2024 IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS), 28-30 Nov. 2024, doi: 10.1109/IoTaIS64014.2024.10799339.

[20] E. Odat and Q. M. Yaseen, "A Novel Machine Learning Approach for Android Malware Detection Based on the Co-Existence of Features," IEEE Access, vol. 11, pp. 15471-15484, 13 Feb. 2023, doi: 10.1109/ACCESS.2023.3244656.

Citation of this Article:

Shaik Salam, & Kalluru Pavankumar Reddy. (2025). Android Malware Detection Using Machine Learning Techniques. In proceeding of International Conference on Sustainable Practices and Innovations in Research and Engineering (INSPIRE'25), published *International Research Journal of Innovations in Engineering and Technology - IRJIET*, Volume 9, Special Issue of INSPIRE'25, pp 24-32. Article DOI https://doi.org/10.47001/IRJIET/2025.INSPIRE04

****

