Android Malware Detection System using Machine Learning

Amanpreet Kaur

Department of Computer Science and Engineering & IT, Jaypee Institute of Information Technology, India,
amanpreet.kaur1410@gmail.com

Sangeeta Lal

School of computing and mathematics, Keele University, Newcastle-under-Lyme, United Kingdom,
s.sangeeta@keele.ac.uk

Shruti Goel

Department of Computer Science and Engineering & IT, Jaypee Institute of Information Technology, India,
shrutigoel1101@gmail.com

Mrinal Pandey

Department of Computer Science and Engineering & IT, Jaypee Institute of Information Technology, India,
mrinalpandey0307@gmail.com

Astha Agarwal

Department of Computer Science and Engineering & IT, Jaypee Institute of Information Technology, India,
astha2agarwal5@gmail.com

Detecting Android malware is imperative for safeguarding user privacy, securing data, and preserving device performance. Consequently, numerous studies have underscored the complexities associated with Android malware detection, prompting a multidimensional approach to tackle these challenges effectively. This research leverages machine learning techniques, emphasizing feature extraction, classification algorithms, and both supervised and unsupervised learning methodologies. The exploration begins with in-depth Exploratory Data Analysis (EDA) to gain insights into the dataset, paving the way for informed decision-making. Principal Component Analysis (PCA) is employed for dimensionality reduction, a pivotal step in handling the multivariate nature of the data. The integration of API calls, clustering, and anomaly detection further enriches the model's capability to discern between benign and malicious applications. Crucially, the study delves into the intricacies of sampling, evaluation, and the Confusion Matrix to quantify the model's performance accurately. The utilization of diverse classification algorithms, including Support Vector Machines (SVM), Multi-Layer Perceptrons (MLP), Random Forest, GaussianNB, Decision Tree, and Logistic Regression, underscores the comprehensive nature of the approach. These algorithms collectively contribute to a robust and versatile Android malware detection model capable of adapting to varying threat scenarios. The dataset employed for training and evaluation is sourced from Kaggle, encompassing 29,999 Android applications categorized as benign or malicious based on permissions sought. Current detection methods, deemed resource-intensive and exhaustive, face the challenge of keeping pace with the relentless evolution of new malware strains. This research seeks to address this gap by proposing a sophisticated, machine learning-driven model that not only enhances accuracy but also demonstrates efficiency and adaptability in the face of a dynamic threat landscape.

Additional Keywords and Phrases: Android, malware Detection, Machine Learning, accuracy, precision

ACM Reference Format:

First Author's Name, Initials, and Last Name, Second Author's Name, Initials, and Last Name, and Third Author's Name, Initials, and Last Name. 2018. The Title of the Paper: ACM Conference Proceedings Manuscript Submission Template: This is the subtitle of the paper, this document both explains and embodies the submission format for authors using Word. In Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY. ACM, New York, NY, USA, 10 pages. NOTE: This block will be automatically generated when manuscripts are processed after acceptance.

1 INTRODUCTION

The proliferation of mobile devices, particularly Android smartphones, has given rise to a significant escalation in the threat posed by malware attacks. Since Android has grown to be the most widely used mobile operating system worldwide, hackers are keen to target it[1]. Malicious software applications often remain concealed within seemingly harmless apps, posing a significant risk to users. They can potentially cause harm to the device, such as slowing it down or rendering it unusable, and can also illicitly acquire critical user data, including personal information and login credentials. Traditional security measures, such as antivirus software, have struggled to keep pace with the evolving complexity of these threats. This is due to the rapid development of new malware variants and the ability of attackers to camouflage their activities. Signature-based antivirus systems often fail to detect these new and sophisticated malware strains. As a potential solution to this constantly expanding problem, machine learning has surfaced as a method for anticipating Android malware. Machine learning algorithms can comprehensively analyze various characteristics and attributes of mobile apps, enabling the identification of potentially hazardous behavior that may go undetected by conventional methods[2]. These algorithms can learn from historical data and adapt to evolving malware tactics, making them a valuable tool in the ongoing battle against Android malware[3]. Despite the increasing menace of malware targeting Android devices, a reliable and robust method for detecting malicious applications remains elusive. The escalating sophistication of malware and its ability to evade conventional defenses necessitates innovative solutions. As Android malware continues to evolve, it becomes increasingly challenging to detect and mitigate these threats effectively. This research aims to solve the main challenge of using machine learning techniques to create a model that can effectively categorize mobile apps as malware (1) or benign (0) depending on the permissions they request. Permissions are crucial in determining an app's behavior, as they specify what resources and data an app can access on the device. Malicious apps often request excessive and unnecessary permissions as part of their deceitful practices, and this is where a machine learning model can play a vital role in identifying suspicious behavior.

The significance of the problem lies in the pressing need for a more efficient and adaptive approach to combat the rising tide of Android malware. The inadequacies of existing security measures emphasize the necessity for innovative solutions. Android malware poses a severe threat to both individual users and organizations, as it can lead to data breaches, financial losses, and damage to a user's digital identity. The urgency to address this issue is further highlighted by the increasing number of malicious apps on official app stores and third-party markets. The novelty of this endeavor stems from its focus on utilizing publicly available metadata information for the detection of Android malware. While traditional approaches often rely on signature-based detection or heuristics, this work seeks to leverage machine learning to analyze app metadata and permissions. This novel approach holds the potential to address the problem more effectively than existing methods, as it adapts to new threats and offers a proactive means of identifying malware based on patterns and behaviors rather than static rules.

To address the problem of Android malware detection, our research includes a comprehensive empirical study. This study encompasses a thorough examination and evaluation of Android metadata and permissions as predictors of malware. It involves collecting and analyzing data from real-world Android apps to understand the relationships between specific permissions and the presence of malware. Additionally, our work introduces a machine learning-based malware detection strategy that relies on the analysis of publicly available metadata information. This strategy involves training machine learning algorithms on a labeled dataset, where apps are classified as benign or malicious. The algorithms then use the knowledge gained during training to classify new, unseen apps. The empirical study seeks to provide valuable insights into the effectiveness of this model and assess its potential as a first-stage filter for detecting Android malware. This approach combines theoretical insights with practical, data-driven research to create a more holistic understanding of Android malware detection.

2 LITERATURE SURVEY

Kumar et al, 2022 [4] focuses on the vulnerabilities of the Android platform to malware attacks, specifically concerning the exploitation of the Android overlay feature by malicious apps. It highlights the challenges faced by conventional malware detection techniques, emphasizing the need for more sophisticated and robust approaches. The paper delves into the application of machine learning techniques for Android malware detection, discussing various methodologies and their limitations. It proposes a lightweight ondevice malware detection solution based on wide learning, aiming to address the resource limitations of mobile devices.

Sarah et al,2021[5] focuses on the detection of Android malware, highlighting the limitations of existing solutions, particularly those based on signature comparisons that fail to identify unknown malware. The research aims to provide a more efficient and accurate solution that can detect various types of malware, minimizing the use of time and resources. It emphasizes the need for an automated process that identifies the most significant and common features in malware, achieving higher accuracy with minimal false positives. The researchers in [2] reduce the feature set from 215 to 100 and achieve an impressive 99.5% accuracy using LightGBM, an ensemble method. They assert the superiority of ensemble methods over traditional machine learning algorithms in malware prediction. The authors talk about how they used the Drebin dataset, using the Recursive Feature Elimination (RFE) method for feature selection and eight different machine learning algorithms.They highlight LightGBM's superiority in achieving the highest accuracy and discuss their analysis of feature ranges for predicting Android malware.

The urgent problem of Android malware and the need for efficient detection techniques to protect the system and user security are the main topics of Ma et al., 2019[6]. The work suggests a cutting-edge combo method that makes use of machine learning techniques to detect malware on Android devices. The work presents the difficulties brought about by the widespread presence of malware for Android devices, emphasizing issues such privilege escalation, remote control, tariff theft, and privacy leaks. The study emphasizes how urgent it is to stop malware from spreading throughout Android markets and devices, as Android is the most popular platform for smartphones and other intelligent devices. The study develops a thorough process that includes de-compiling Android applications and creating a control flow graph (CFG) from the source code in order to address these issues. The work then extract Application Program Interface (API) calls from the CFG to build three distinct data sets: Boolean, frequency, and chronological data sets. Using these data sets, the research team constructs three detection models: the API Usage Detection Model, API Frequency Detection Model, and API Sequence Detection Model, utilizing various machine learning techniques. The paper discusses the existing research landscape in the domain of Android malware, citing efforts from both industry and academia. It emphasizes the significance of understanding Android permissions and the limitations of the current security measures. Through extensive experimentation on a large dataset comprising benign applications and malicious samples, the paper demonstrates the efficacy of the proposed approach. While the API Frequency Detection Model detects 97% of malware samples with a false-positive rate of 9.1%, the API Usage Detection Model obtains a 95% detection rate with a 6.2% false-positive rate. With a 2.9% false-positive rate, the API Sequence Detection Model outperforms the others, detecting 99% of malware samples. The significance of their approach is emphasized in the paper, highlighting the novelty of their contributions to the field, especially in the creation of chronological datasets using static detection methods and the application of the Long Short-Term Memory (LSTM) algorithm for building the detection model.

The increasing concern over Android malware is highlighted by Feng et al., 2021[7], as the number of Android devices and apps keeps rising. Users are increasingly storing personal data on their mobile devices through various apps, making security and privacy a top priority. Android malware poses a significant security threat, and traditional server-side malware detection solutions have limitations, especially when dealing with apps from unofficial markets and thirdparty resources. The paper emphasizes the need for a last line of defense on mobile devices and proposes an effective solution called MobiTive. MobiTive is a pre-installed solution that uses specialized deep neural networks for responsive and real-time malware detection on mobile devices, in contrast to typical app scanning and monitoring engines. By taking into account manifest properties and API calls from Dalvik binary files, enhancing feature inputs, and contrasting several deep neural network models, the study expands on earlier findings. The system's effectiveness is demonstrated through multi-class classification tasks, usability evaluations on real mobile devices, and performance comparisons with existing solutions. The findings show that MobiTive is a mobile malware detection solution for Android that is both effective and efficient, attaining a high classification accuracy of 96.78% with little overhead. Overall, the research presents MobiTive as a practical and secure last line of defense against Android malware on mobile devices, addressing the evolving landscape of security threats in the mobile app ecosystem.

Nasri et al.,2020[8] aims to address the increasing threat of malware targeting Android operating systems. The abstract outlines the importance of Android malware detection to prevent more serious security issues. It introduces two primary methods of analysis: static analysis, which examines an application's code meticulously, and dynamic analysis, which monitors an application's behavior to identify malware. The paper proposes a malware detection system using a machine learning approach, with a focus on Android operating systems. The research uses a dataset comprising 10,000 samples of malware and 10,000 benign applications. It employs five different classifiers, and the results indicate that the Random Forest classifier achieved the highest accuracy at 89.36%, slightly outperforming Naïve Bayes at 89.2%. The paper emphasizes the importance of true positive rate (TPR) for accurately predicting malware processes and false positive rate (FPR) for wrongly classifying normal applications as malware. The evaluation employs the area under the curve (AUC) to determine the overall detection accuracy. Additionally, it highlights that Naïve Bayes has the advantage of lower model complexity, making it a faster option for building the model. The paper cites a Symantec report from 2018, which notes a 10% increase in targeted activities and a significant number of devices infected with RottenSys malware. This situation underscores the need for effective malware detection methods to protect Android smartphones, given their widespread use for activities like online shopping, banking, and cloud storage.

Mahindru et al., 2017[9] focuses on how the open nature and widespread use of the operating system make Android-based apps vulnerable to malware assaults. In addition to presenting the findings of an analysis performed on a dataset consisting of 11,000 Android application packages (.apk), the study suggests a dynamic analytic approach for detecting malware in Android applications. The paper's contributions include the extraction of a set of 123 dynamic permissions from the dataset and the evaluation of various machine learning classification techniques for detecting malicious Android applications. The paper specifically opts for a dynamic analysis approach due to its effectiveness against malware obfuscation. The introduction highlights the dataset construction process, which involves the categorization of applications into various domains and the classification of these applications as normal or malware.

Kouliaridis et al.,2021[10] discusses the growing threat of mobile malware attacks, particularly on the Android platform. It highlights that existing mobile malware detection solutions rely on machine learning, but there is a lack of standardization in terms of metrics, models, datasets, and analysis techniques used in the field. This lack of standardization hinders the cross-comparison of detection schemes and raises questions about the reliability of results. To address this issue, the paper aims to organize ML-powered malware detection approaches from the last seven years into four axes: dataset age, analysis type, ML techniques, and performance metrics. Additionally, it presents a convergent framework that will serve as a foundation for machine learning processes in the industry and direct future methods for detecting Android malware.

McLaughlin et al.,2017[11] introduces a novel Android malware detection system that leverages a deep convolutional neural network (CNN) for classification based on the static analysis of raw opcode sequences obtained from disassembled programs. Unlike traditional malware detection methods that rely on manually designing signatures, this system automatically learns features indicative of malware from the opcode sequences, eliminating the need for hand-crafted features. By teaching the network end-to-end to acquire relevant characteristics and carry out classification, the suggested technique streamlines the training procedure and does away with the requirement to count millions of n-grams during training. The system's computational efficiency is emphasized, with training and testing times scaling linearly with the number of malware examples. It can be executed on GPUs, which are standard in many mobile devices, allowing for the rapid scanning of malware files. The paper anticipates that as more training data becomes available, the system's accuracy will improve due to the high learning capacity of neural networks. Furthermore, the proposed method draws inspiration from existing n-gram-based approaches. The convolutional network inherently learns to detect n-gram-like signatures by identifying sequences of opcodes indicative of malware. Furthermore, the method is able to find very long n-gram-like signatures, for which explicit enumeration would be problematic. The automated system's malware signatures may complement those found by human analysts, making it valuable for use alongside conventional malware signature databases.

The research collectively emphasizes the increasing threat of Android malware and the pressing need for sophisticated, robust, and efficient detection systems. The researchers highlight the challenges posed by the open nature of the Android platform and the constant evolution of malware, necessitating the adoption of innovative approaches. Machine learning emerges as a key tool in combating Android malware, with various studies focusing on its application to enhance detection accuracy.

In order to overcome this limitation, our work’s objective is to create a robust machine learning model capable of accurately distinguishing Android malware from benign apps. This model will rely on the behavioral traits, coding characteristics, and other relevant features of Android applications. The project aims to offer a powerful tool for enhancing Android malware detection, leveraging machine learning for efficient classification.

3 PROPOSED SOLUTION

3.1 Data Preprocessing

Data collection begins by importing the dataset from a CSV file. The dataset used for model training and assessment is sourced from Kaggle and contains information on permissions for around 30,000 Android apps. This dataset includes 183 characteristics, such as Dangerous Permissions Count, Default: Access DRM material, Default: Transfer Application Resource, etc. The dataset features a binary target class, "Class," which distinguishes between benign (binary 0) and malicious (binary 1) apps.

In total, the dataset comprises 29,999 records, with 9,999 categorized as benign apps and 20,000 as malicious apps.

The dataset serves as the foundation for training and testing the machine learning model. The dataset is structured and manipulated for ease of use by loading it into a data frame, facilitating efficient data handling. Necessary attributes relevant to Android app behavior, coding characteristics, and other distinguishing features are extracted from the dataset. These attributes are crucial for the accurate classification of benign and malicious apps. Data quality is ensured by identifying and addressing any missing values within the dataset. Specifically, missing values are imputed by replacing them with the mean of the corresponding column. This step is essential to create a complete and reliable dataset for analysis.

Data Visualization and Analysis: To gain a deeper understanding of the dataset, the project proceeds with an indepth data analysis, further exploring the distribution and characteristics of the data. Visualizations are generated using libraries like Matplotlib and Seaborn. These visual representations may include histograms, scatter plots, and other graphical elements. Visualizations aid in uncovering patterns, outliers, and potential disparities between benign and malicious apps. Data visualization and analysis play a crucial role in identifying trends, patterns, and areas of interest within the dataset. This visual insight helps guide subsequent analysis.

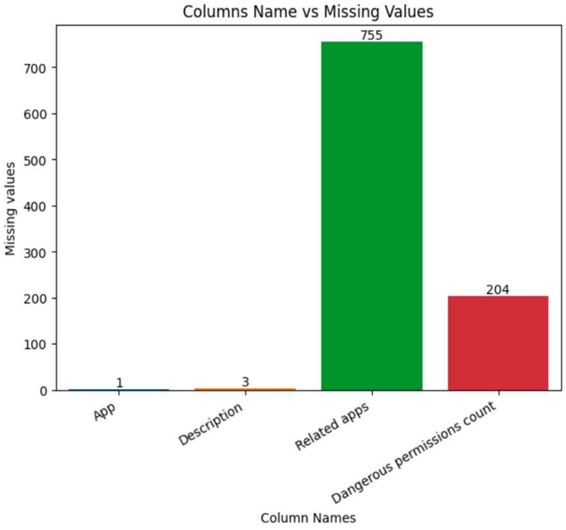


Figure 1 : Column Name vs Missing Values

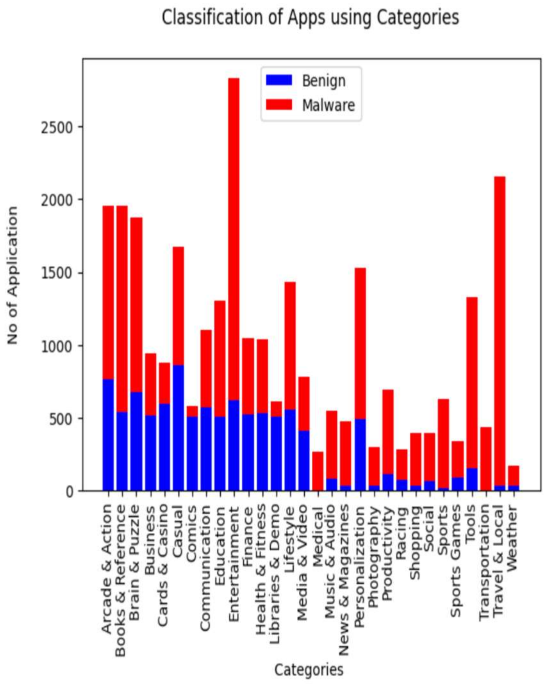


Figure 2: Classification of Apps using Categories

3.2 Exploratory Data Analysis (EDA)

EDA is a pivotal phase that involves a detailed investigation into the dataset's attributes and their relationships[12]. The objective is to identify the most influential predictors for the classification of Android apps. During EDA, a range of statistical and visual techniques are employed to unearth patterns and correlations within the dataset[13]. For example, correlations between app permissions, coding characteristics, and app behavior may be explored. The insights obtained through EDA serve as a foundation for subsequent feature selection and model development. By recognizing which attributes are most significant for distinguishing between benign and malicious apps, EDA plays a vital role in guiding the modeling process.

3.3 Methodology

Following data preprocessing and EDA, the dataset is split into training and testing sets. This division ensures the model's performance can be assessed on unseen data, contributing to the model's reliability. Multiple machine learning classifiers are applied to the dataset, including logistic regression, decision trees, and Naive Bayes. Despite these initial attempts, the results may fall short of the desired accuracy. To address the complexities inherent in the dataset, Principal Component Analysis (PCA) is introduced. PCA serves to reduce the dimensionality of the data, which is especially valuable when dealing with multivariate data tables. The variance percentage is examined to determine the optimal number of principal components to retain. An inverse transform is employed to reconstruct the data post-dimensionality reduction. The Random Forest classifier is integrated into the dataset, marking a significant improvement in prediction accuracy and model performance. Boosting techniques are subsequently applied to further enhance the model's predictive accuracy. These techniques are employed both on unsampled data and on data with selected reliable features. Support Vector Machines (SVM) and Multi-Layer Perceptrons (MLP) are leveraged to achieve the best results, signaling substantial progress and achieving high accuracy[14][15]. When comparing the results obtained after feature selection and boosting, it becomes evident that significant progress has been made. The final model demonstrates a high level of accuracy. The comprehensive solution approach outlined above is designed to achieve the project's primary objectives, which include creating an effective machine learning model for Android malware detection. This model not only enhances detection accuracy but also serves as a valuable tool for security researchers and developers in the Android app ecosystem. The ultimate measure of the model's success is its final accuracy in classifying Android malware, a testament to its effectiveness in bolstering mobile device security.

4 RESULTS

Precision: All models have similar precision values (around 0.67-0.68), indicating that they are comparable in terms of positive prediction accuracy.

Accuracy: Again, all models have similar accuracy values (0.67-0.68), showing consistency in overall prediction accuracy.

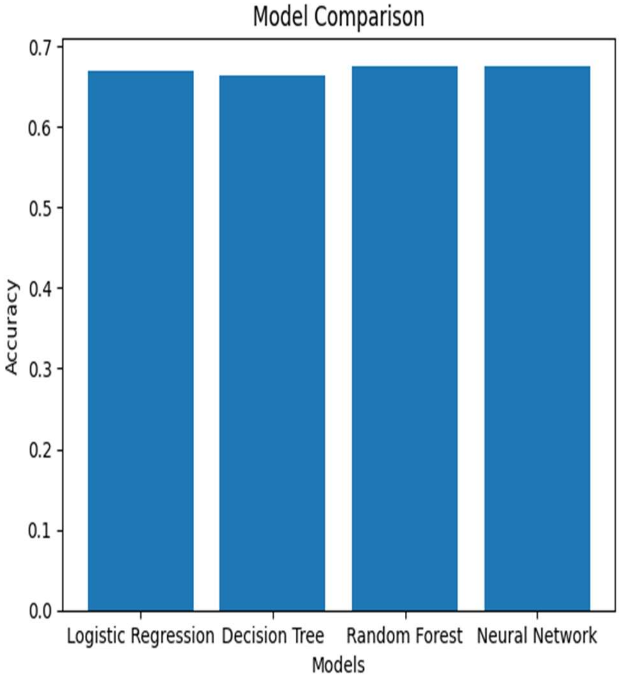
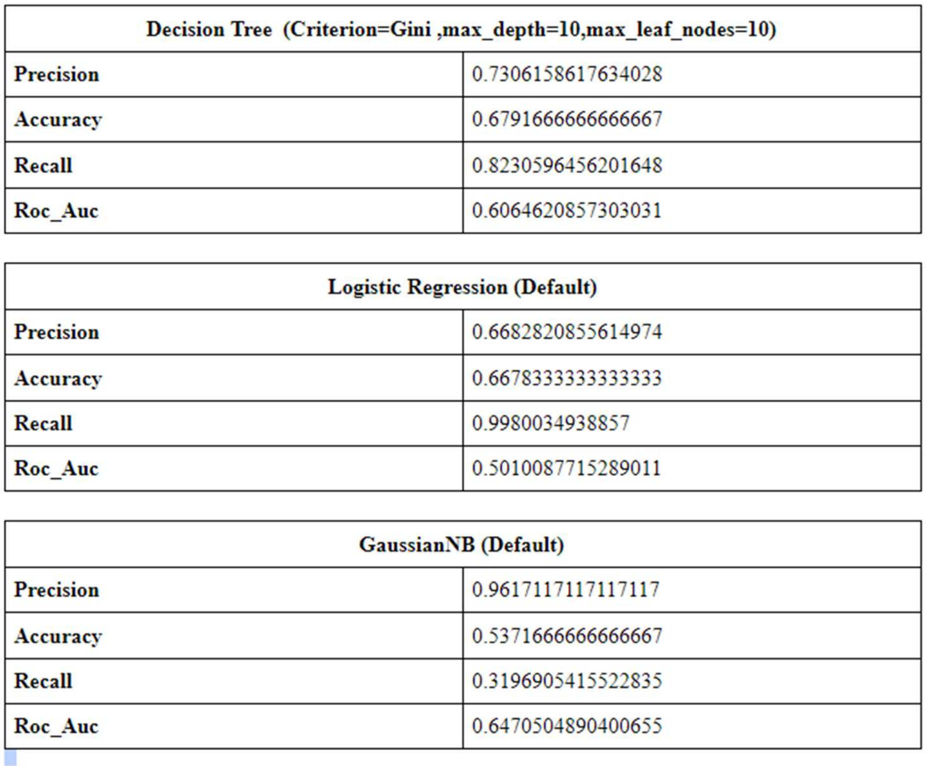


Figure 3: Comparison of models in terms of accuracy

Recall: The recall for the Logistic Regression model is slightly lower (0.96), indicating that it may miss a few relevant instances compared to Naive Bayes and Decision Tree models.

ROC Score: In comparison to Naive Bayes and Decision Tree, Logistic Regression has a little higher ROC-AUC score (0.53), suggesting a better trade-off between true positive and false positive rates. An improved trade-off between true positive and false positive rates is shown by a higher ROC-AUC score. The probability that a randomly selected positive instance will have a greater predicted probability than a randomly selected negative instance is what is known as the ROC-AUC score, which is computed based on the ROC curve.



Decision Tree (Criterion=Gini ,max_depth=10,max_leaf_nodes=10)	
Precision	0.7306158617634028
Accuracy	0.6791666666666667
Recall	0.8230596456201648
Roc_Auc	0.6064620857303031

Logistic Regression (Default)	
Precision	0.6682820855614974
Accuracy	0.6678333333333333
Recall	0.9980034938857
Roc_Auc	0.5010087715289011

GaussianNB (Default)	
Precision	0.9617117117117117
Accuracy	0.5371666666666667
Recall	0.3196905415522835
Roc_Auc	0.6470504890400655

Figure 4 : Comparison in terms of ROC Curve

5 CONCLUSION

The research presented in this work has significant implications for Android malware detection and mobile security. A machine learning model was successfully developed to accurately predict whether an Android app is malware or benign. The model's high precision and recall rates make it a valuable tool for security researchers and developers. The trained machine learning model represents a significant deliverable, demonstrating its effectiveness in classifying Android apps with high accuracy. The methodology used in this study combines data collection, preprocessing, and machine learning to create an efficient solution for Android malware detection. The research has the potential to improve Android malware detection in several ways, including better detection, real-time protection, fewer false positives, and enhanced user experience. It can be integrated into software solutions to provide the benefits of improved security and user experience.

The research paves the way for future work and improvements in the domain of Android malware detection. Future work can focus on incorporating additional features and data sources to further improve the model's accuracy and adaptability.

Also, The development of real-time malware detection tools that can continuously monitor and protect Android devices is a promising avenue for future research. Ongoing research can aim to refine the model to reduce the occurrence of false positives, enhancing the precision of malware detection. Future research can explore ways to enhance the overall user experience by detecting and blocking malware in real time, ensuring users feel more secure while using their devices. Research can continue to develop models that swiftly update their detection skills to respond to new threats, thereby lowering the overall risk of malware assaultsThe research's findings and the developed model can be integrated into software solutions for broader use in the Android ecosystem.

In conclusion, this work not only provides a valuable solution for Android malware detection but also opens the door to a multitude of potential future research endeavors that can further enhance mobile security and user experience in the Android ecosystem.

REFERENCES

[1] Al-Ofeishat, H. A. (2024). Enhancing Android Security: Network-Driven Machine Learning Approach For Malware Detection. Journal of Theoretical and Applied Information Technology, 102(2), 737-750.

[2] Jyothish, A., Mathew, A., & Vinod, P. (2024). Effectiveness of machine learning based android malware detectors against adversarial attacks. Cluster Computing, 27(3), 2549-2569.

[3] Alhamri, R. Z., Cinderatama, T. A., Eliyen, K., & Izzah, A. (2024). Supervised Learning Methods Comparison for Android Malware Detection Based on System Calls Referring to ARM (32-bit/EABI) Table. Journal of Information Technology and Cyber Security.

[4] Hareram Kumar and Prof. Sarwesh Site. "Android Malware Prediction using Machine Learning Techniques: A Review." International Journal of Recent Development in Engineering and Technology, vol. 11, no. 2, Feb. 2022.

[5] Neamat Al Sarah, Fahmida Yasmin Rifat, Md. Shohrab Hossain, Husnu S. Narman. An Efficient Android Malware Prediction Using Ensemble machine learning algorithms. Procedia Computer Science, Volume 191, 2021, Pages 184-191. doi:10.1016/j.procs.2021.07.023

[6] Z. Ma, H. Ge, Y. Liu, M. Zhao and J. Ma, "A Combination Method for Android Malware Detection Based on Control Flow Graphs and Machine Learning Algorithms," in IEEE Access, vol. 7, pp. 21235-21245, 2019, doi: 10.1109/ACCESS.2019.2896003.

[7] R. Feng, S. Chen, X. Xie, G. Meng, S. -W. Lin and Y. Liu, "A Performance-Sensitive Malware Detection System Using Deep Learning on Mobile Devices," in IEEE Transactions on Information Forensics and Security, vol. 16, pp. 1563-1578, 2021, doi: 10.1109/TIFS.2020.3025436.

[8] Nasri, Nuren Natasha Maulat, Mohd Faizal Ab Razak, RD Rohmat Saedudin, Salwana Mohamad, and Ahmad Firdaus Asmara. "Android malware detection system using machine learning." International Journal 9, no. 1.5 (2020).

[9] Arvind Mahindru and Paramvir Singh. 2017. Dynamic Permissions based Android Malware Detection using Machine Learning Techniques. In Proceedings of the 10th Innovations in Software Engineering Conference (ISEC '17). Association for Computing Machinery, New York, NY, USA, 202–210. https://doi.org/10.1145/3021460.3021485.

[10] Kouliaridis V, Kambourakis G. A Comprehensive Survey on Machine Learning Techniques for Android Malware Detection. Information. 2021; 12(5):185. https://doi.org/10.3390/info12050185.

[11] Niall McLaughlin, Jesus Martinez del Rincon, BooJoong Kang, Suleiman Yerima, Paul Miller, Sakir Sezer, Yeganeh Safaei, Erik Trickel, Ziming Zhao, Adam Doupé, and Gail Joon Ahn. 2017. Deep Android Malware Detection. In Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy (CODASPY '17). Association for Computing Machinery, New York, NY, USA, 301–308. https://doi.org/10.1145/3029806.3029823

[12] Smmarwar, S. K., Gupta, G. P., & Kumar, S. (2024). Android Malware Detection and Identification Frameworks by Leveraging the Machine and Deep Learning Techniques: A Comprehensive Review. Telematics and Informatics Reports, 100130..

[13] Pathak, A., Barman, U., & Kumar, T. S. (2024). Machine learning approach to detect android malware using feature-selection based on feature importance score. Journal of Engineering Research.

[14] Misalkar, H., & Harshavardhanan, P. (2024). Assessing the efficacy of Machine learning classifier for Android malware detection. Journal of Integrated Science and Technology, 12(4), 788-788.

[15] Xie, W., & Zhang, X. (2024, January). The Application of Machine Learning in Android Malware Detection. In 2024 4th International Conference on Neural Networks, Information and Communication (NNICE) (pp. 1-4). IEEE.