IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 11, Issue 9, September 2024

DOI: 10.17148/IARJSET.2024.11906

Android Malware Detection Through ML-Based Analysis Of APK Permissions

Sairaj Paygude[1], Sonal Sonawane[2], Siddham Tatiya[3], Nakul Sarda[4], Ms. A. Dirgule[5]

Student, Computer Engineering, Sinhgad College of Engineering, Pune, India[1-4]

Assistant Professor, Computer Engineering, Sinhgad College of Engineering, Pune, India[5]

Abstract: The exponential growth of Android-based devices has resulted in a worrying rise in the spread of malware through mobile applications. The surge in Android malware highlights the crucial need for strong security measures. Machine learning, focusing on APK permission analysis, offers a promising solution to detect harmful apps and protect users from security threats and privacy breaches. The model classifies the APK files as benign or malicious based on the permissions used by the model. Our paper, primarily research-based, focuses on comparing various available options for detecting malware to identify the most suitable real-time solution. We propose a malware detection system that assesses an app's maliciousness by analyzing its permission usage. This study presents an innovative method for detecting Android malware, employing Support Vector Machines (SVM), as the machine learning model of choice after evaluating other models. In addition to comparing various models, we incorporated feature reduction techniques during the assessment process. After a comprehensive comparison of various parameters among different models, Support Vector Machines (SVM) emerged as the most suitable choice for our research. The feasibility of SVM was determined through measures such as ROC-AUC, recall, precision, accuracy, and F1-score.

Keywords: Machine learning, APK, Malware, Android, permissions.

I. INTRODUCTION

In response to the escalating threat of Android malware, the proposed "Application for Android Malware Detection through ML-Based Analysis of APK Permissions" leverages machine learning to safeguard users. By scrutinizing APK permissions, it identifies potential security risks, utilizing algorithms trained on extensive datasets to classify apps as benign or malicious. The framework entails decompiling APKs, extracting permissions, and employing machine learning models for classification. The paper discusses related work, the approach, experimented machine learning algorithms, conducted experiments, and concludes with findings and avenues for future research.

II. RELATED WORK

Starting from a foundational standpoint, our previous work has already delved into the extensive landscape of Android malware detection. In our prior publication, we conducted a comprehensive literature survey, exploring seminal studies and emerging trends in this domain. Building upon this groundwork, the current literature review further expands our understanding by encompassing additional research contributions and advancements in the field.[11]

The literature review encompasses several seminal studies in the realm of Android malware detection. Akbar, Hussain, Mumtaz, Riaz, Wahab, and Jung [1] introduce PerDRaML, a permissions-based malware detection system utilizing machine learning techniques to achieve high accuracy. Elayan and Mustafa [2] propose a deep learning approach for Android malware detection, showcasing the superiority of deep learning over traditional methods. Al and Mouheb [3] address smartphone vulnerability to cyber-attacks by advocating for a machine learning approach focused on static features. Agrawal and Trivedi [4] evaluate machine learning classifiers for Android malware detection, emphasizing the efficacy of machine learning methods.

Sandeep HR [5] presents a static analysis approach for Android malware detection using deep learning, highlighting the importance of feature extraction from APK files. Alswaina and Elleithy [6] conduct a comprehensive survey on Android malware family classification and analysis, advocating for advanced artificial intelligence and big data technologies. Agrawal, Shah, Sonam Chavan, Ganesh Gourshete, and Shaikh [7] propose a system integrating machine learning algorithms and semantic analysis to enhance malware detection. Sabbah Birzeit, Taweel Birzeit, and Zein [8] provide a comprehensive literature review on Android malware detection, shedding light on prevalent techniques and challenges in the field.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 11, Issue 9, September 2024

DOI: 10.17148/IARJSET.2024.11906

Wang and Yan [9] propose an innovative malware detection method using text semantics of network flows, showcasing high accuracy and practical applicability. Razgallah, Khoury, Hallé, and Khanmohammadi [10] offer an extensive survey of malware detection techniques in Android apps, providing recommendations for future research to combat evolving threats effectively. These studies collectively underscore the significance of machine learning, deep learning, and semantic analysis in addressing the escalating threat of Android malware, while also advocating for standardized datasets and automated detection methods for enhanced security measures in Android applications.

III. METHODOLOGY

A. Tools and Technologies Used:
In our research, we utilized APK Tool, a potential tool for analysing Android application packages (APKs), to extract permissions from the apps under investigation. After conducting thorough research on available options like Androguard, we selected APK Tool due to its superior suitability for our approach. This tool provided us with comprehensive insights into the permissions requested by each app, enabling us to understand their potential functionalities and security implications.

B. Importance of APK Permissions:
APK permissions serve as crucial access rights granted to Android applications upon installation. They govern the actions an app can perform on a device, encompassing functionalities such as accessing contacts, camera, and location. In the context of malware detection, the significance of permissions cannot be overstated. Malicious applications often exploit excessive permissions to access sensitive user data or execute harmful actions without user consent. By scrutinizing these permissions, our approach aims to identify suspicious behavior indicative of malware presence, thereby enhancing our ability to accurately classify apps as benign or malicious.

C. Training and Testing:
To assess the efficacy of our approach, we conducted experiments on a dataset comprising 29,332 records, with each record consisting of 86 static features (attributes) [1]. Thee dataset was divided into a training set, comprising 75% of the data, and a testing set, comprising the remaining 25%. During static analysis, data regarding the permissions accessed by the app was collected while the app remained static. This approach proved to be the fastest and most costeffective, as it did not require executing the application or monitoring activities.[6]

D. Machine Learning Algorithms:
Our experiments aimed to evaluate the performance of various machine learning algorithms in distinguishing between benign and malicious APKs. The algorithms employed include Support Vector Machine (SVM), K-Nearest Neighbours (KNN), Decision Tree (DT), Random Forest (RF) ,Logistic Regression (LR), Naive Bayes (NB), and Artificial Neural Network (ANN).

E. Feature Reduction Techniques:
Additionally, we employed feature reduction techniques to enhance the efficiency of our analysis. Specifically, we utilized Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA) to reduce the dimensionality of our dataset and improve the computational efficiency of our models. These techniques aid in extracting the most relevant features from the dataset, facilitating more accurate and efficient malware detection [1].

Principal Component Analysis (PCA) is a type of unsupervised dimensionality reduction technique which takes into account variance preservation, useful for exploratory data analysis and visualization. It identifies orthogonal axes, called principal components, to transform high-dimensional data into a lower-dimensional space. Linear Discriminant Analysis (LDA), in contrast, is a supervised technique focusing on maximizing class separability during dimensionality reduction. It considers class labels to identify linear discriminants that best separate different classes, making it particularly suitable for classification tasks.

IV. ANALYSIS

After conducting experiments with multiple machine learning methods, we obtained various results that were analysed based on several evaluation parameters. This section provides a summary of the evaluation metrics utilized to gauge the effectiveness of the machine learning algorithms applied in our study on Android malware detection via ML-based analysis of APK permissions.

The assessment criteria includes accuracy, recall, precision, F1-score, and the area under the ROC curve (ROCAUC). These all metrics provide insights into the effectiveness of the machine learning models in distinguishing between benign and malicious APKs.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 11, Issue 9, September 2024

DOI: 10.17148/IARJSET.2024.11906

Accuracy: Accuracy reflects how often the machine learning models correctly classify instances. It's a crucial metric indicating the overall effectiveness of the models in distinguishing between benign and malicious APKs. Higher accuracy means better classification performance, enhancing Android device security.

Precision: Precision evaluate the proportion of truely identified malicious instances among all instances classified as malicious by the models. It's essential for minimizing false alarms or misclassifications of benign apps as malicious. Higher precision indicates fewer false positives, ensuring users aren't needlessly alerted about their app safety.

Recall: Recall, or sensitivity, gauges the proportion of truely identified malicious instances of all the actual malicious ones. It's critical for capturing all instances of malware, reducing the risk of overlooking security threats. Higher recall means better detection accuracy of malicious apps.

F1-score: F1-score harmonizes precision and recall, offering a balanced measure of model performance. It provides a single value reflecting the overall effectiveness in both false negatives and false positives. Higher F1-score indicates a better balance between recall and precision, ensuring robust performance in classifying benign and malicious apps.

ROC-AUC Score: The ROC-AUC score assesses how correctly the model can differentiate among benign and malicious apps at different thresholds. Higher scores signify better discrimination between the two classes, indicating superior performance in differentiating between safe and harmful apps, thus aiding in assessing security capabilities.

By analysing these evaluation parameters, we gain a comprehensive understanding of the strengths and weaknesses of each machine learning method in detecting Android malware based on APK permissions. This analysis enables us to identify the most effective approach for mitigating the threat of malicious applications on Android devices.

V. PERFORMANCE RESULTS:

After thoroughly evaluating the performance results across different machine learning models and feature reduction techniques, it becomes evident that the Support Vector Machine (SVM) model outshines its counterparts in terms of accuracy and time for Android malware detection. Among the models tested, including Decision Tree, Random Forest, Naive Bayes, Logistic Regression, K-Nearest Neighbours (KNN), and Artificial Neural Network (ANN), SVM consistently demonstrates superior performance. Notably, SVM exhibits the highest accuracy, precision, recall, F1-score, and ROC-AUC score across multiple scenarios.

TABLE I. MODELS WITHOUT FEATURE REDUCTION

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC	Time
DT	0.962	0.960	0.963	0.962	0.962	0.001
RF	0.968	0.966	0.970	0.968	0.968	0.016
NB	0.606	0.995	0.558	0.715	0.767	0.001
LR	0.957	0.947	0.953	0.957	0.957	0.001
KNN	0.960	0.963	0.956	0.959	0.960	0.024
SVM	**0.969**	**0.970**	**0.967**	**0.968**	**0.968**	**0.003**
ANN	0.965	0.955	0.974	0.964	0.965	0.014

TABLE II. MODELS USING PCA FEATURE REDUCTION

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC	Time
DT	0.956	0.950	0.961	0.956	0.956	0.001
RF	0.963	0.960	0.966	0.963	0.963	0.011
NB	0.920	0.893	0.942	0.917	0.921	0.001
LR	0.945	0.947	0.943	0.945	0.945	0.001
KNN	0.957	0.960	0.955	0.957	0.957	0.007
SVM	**0.968**	**0.969**	**0.966**	**0.968**	**0.968**	**0.002**

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 11, Issue 9, September 2024

DOI: 10.17148/IARJSET.2024.11906

TABLE III. MODELS USING LDA FEATURE REDUCTION

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC	Time
DT	0.957	0.954	0.959	0.956	0.957	0.001
RF	0.957	0.955	0.959	0.957	0.957	0.010
NB	0.948	0.963	0.934	0.949	0.949	0.001
LR	0.949	0.963	0.935	0.949	0.949	0.001
KNN	0.956	0.951	0.960	0.955	0.956	0.010
SVM	**0.958**	**0.970**	**0.946**	**0.958**	**0.958**	**0.001**

Feature reduction techniques like Principal Component Analysis (PCA) and also Linear Discriminant Analysis (LDA) are crucial for optimizing machine learning models in Android malware detection. They enhance computational efficiency and prevent overfitting by reducing the dimensionality of the dataset. Integrating PCA and LDA into the SVM model improves accuracy and ensures scalability in real-world deployment. In summary, these techniques are essential for maximizing the effectiveness of malware detection systems[4].

For instance, when considering the baseline scenario without feature reduction, SVM achieves an accuracy of 96.86%, outperforming other models such as Decision Tree (95.62%), Random Forest (96.33%), Naive Bayes (60.64%), Logistic Regression (95.74%), KNN (95.96%), and ANN (96.45%). Furthermore, SVM maintains competitive accuracy levels even when feature reduction techniques like Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are implemented.

Although SVM with LDA provides less time overhead, SVM with PCA yields higher accuracy. The time difference between LDA and PCA is considerably small, with 0.00186 and 0.00134, respectively. Similarly, standard SVM, despite its high accuracy, takes more time than both variants of SVM.

Apart from demonstrating superior accuracy, Support Vector Machine (SVM) with PCA analysis exhibits efficient computational performance, with relatively low execution times compared to alternative models. While some models may achieve comparable accuracy, they often require longer processing times, rendering them less practical for real-time malware detection applications. The effectiveness of SVM can be attributed to its proficiency in classifying complex and high-dimensional data, making it well-suited for distinguishing between benign and malicious APKs.

Additionally, SVM's robustness to overfitting and its capability to handle both linear and non-linear data separation contribute to its superior performance in this context. Overall, analysing these insights highlights that SVM not only achieves the highest accuracy but also provides a balance between precision, recall, and time efficiency, making it the most suitable choice for Android malware detection.

Furthermore, SVM's ability to generalize well to new data and handle high-dimensional feature spaces further reinforces its superiority over alternative models in this domain.

Visualizations are provided for further clarification and understanding, aiding in comprehending the performance metrics and comparisons between different machine learning models.

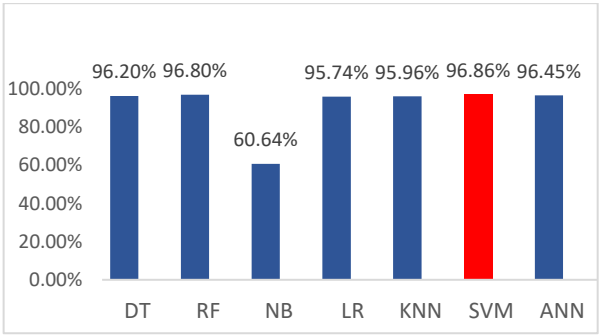

Fig. 1. Accuracies of models without feature reduction.

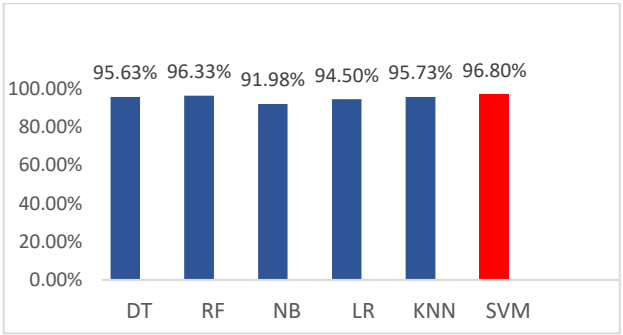

Fig.2. Accuracies of models using PCA feature reduction technique.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 11, Issue 9, September 2024

DOI: 10.17148/IARJSET.2024.11906

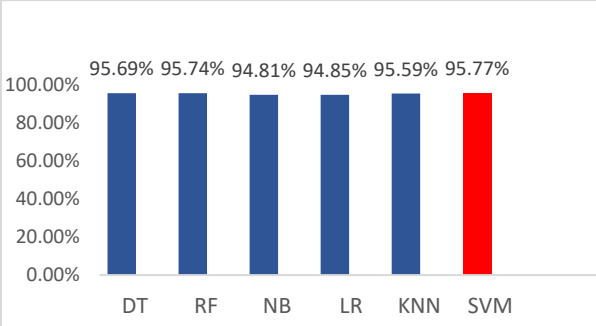

Fig. 3.Accuracies of models using LDA feature reduction technique.

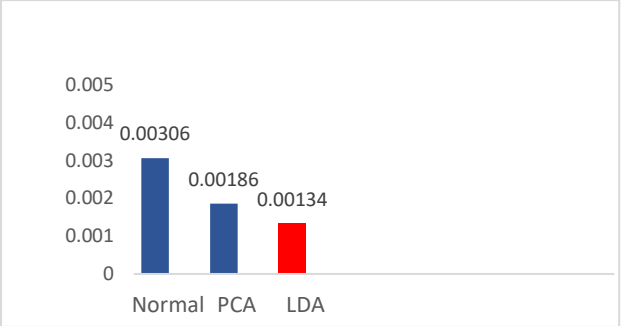

Fig. 4. Time comparison between above three SVM models.

VI. CONCLUSION

In conclusion, this research paper focuses on Android malware detection through machine learning based analysis of static APK permissions. With the proliferation of Android devices and the escalating threat of malware, robust security measures are imperative. Through machine learning, particularly the Support Vector Machine (SVM) model, a novel approach is proposed to detect and classify Android applications as benign or malicious based on their permission usage. The methodology involves extracting permissions from APK files using APK Tool, investigating various machine learning algorithms, and evaluating feature reduction techniques such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). The experiments conducted on a comprehensive static dataset reveal SVM's superiority in accuracy, precision, recall, and time efficiency. SVM emerges as the most effective model for Android malware detection, accurately classifying complex data while maintaining efficient processing. Scrutinizing permissions is crucial for malware detection, enabling the identification of suspicious behavior and mitigation of security threats.

ACKNOWLEDGMENT

We extend our deepest appreciation to all contributors and supporters who have enriched this research endeavor with their invaluable insights and unwavering encouragement.

REFERENCES

[1]. F. Akbar, M. Hussain, R. Mumtaz, Q. Riaz, A.W.A. Wahab, K.-H. Jung, "Permissions based detection of android malware using machine learning," Symmetry, 2022.
[2]. N. Elayan, A.M. Mustafa, "Android malware detection using deep learning," in Proceedings of the 2nd International workshop on Data-Driven Security (DDSW 2021), March 23 - 26, 2021.
[3]. Ali Al, D. Mouheb, "Android Malware Detection Using Static Features And Machine Learning," IEEE, 2020.
[4]. P. Agrawal, B. Trivedi, "Evaluating machine learning classifiers to detect android malware," in Proceedings of the IEEE International Conference for Innovation in Technology (INOCON), Bengaluru, India, Nov 6-8, 2020.
[5]. S. HR, "Static analysis of android malware detection using deep learning," in Proceedings of the International Conference on Intelligent Computing and Control Systems (ICICCS), 2019.
[6]. F. Alswaina, K. Elleithy, "Android malware family classification and analysis: current status and future directions," Electronics, 2020.
[7]. R. Agrawal, V. Shah, S. Chavan, G. Gourshete, N. Shaikh, "Android malware detection using machine learning," in Proceedings of the International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), 2020.
[8]. A. Sabbah Birzeit, A. Taweel Birzeit, S. Zein, "Android malware detection: a literature review," Communications in Computer and Information Science, February 2023.
[9]. S. Wang, Q. Yan, "Detecting android malware leveraging text semantics of network flows," IEEE Transactions on Information Forensics and Security, vol. 13, no. 5, May 2018.
[10]. A. Razgallah, R. Khoury, S. Hallé, K. Khanmohammadi, "A survey of malware detection in Android apps: Recommendations and perspectives for future research," Computer Science Review, 2021.
[11]. S. Paygude, S. Sonawane, S. Tatiya, N. Sarda, "Literature survey on android malware detection through ml - based analysis," International Advanced Research Journal in Science, Engineering and Technology, vol. 10, issue 10, October 2023.

