JOURNAL OF THE ROYAL LAUREATES ACADEMY

www.rlaindia.org

A COMPARATIVE STUDY OF MACHINE LEARNING MODELS FOR PERMISSION-BASED ANDROID MALWARE DETECTION

Hemender Kumar

Research Scholar, Department Of Computer Science, Shri Venkateshwara University, Gajraula, Amroha, (Uttar Pradesh)

Dr. Rakesh Kumar Yadav

Professor, Shri Venkateshwara University, Gajraula, Amroha, (Uttar Pradesh)

ABSTRACT

Smartphones have grown in sophistication and appeal in recent years. A proliferation of malware specifically targeting smartphones has coincided with the device's meteoric growth in popularity and the abundance of personally identifiable information stored on them. This research looks into how well permission-based features work with supervised machine learning to differentiate between safe and dangerous Android apps. In order to guarantee that the samples were safe, we used the AndroZoo repository to gather APK files, specifically those from Google Play apps. The five machine learning classifiers that were trained and tested with tenfold cross-validation are Random Forest, J48, Multi-Layer Perceptron, Decision Table, and Naïve Bayes. Accuracy, precision, recall, false positive rate, and F-measure were used to evaluate performance. With an accuracy of 89.40% and the lowest false positive rate, Random Forest outperformed other models in terms of overall performance, dependability, and consistency, according to the results. The use of confusion matrices provided more evidence that tree-based models are better at detecting malware with lower mistake rates.

Keywords: Android, Machine Learning, Malware, Supervised, Permission.

I. INTRODUCTION

Smartphones have become an integral part of our daily lives and play a big role in mobile security due to the integration of technology. We must stand united in our opposition to the spread of Android malware. Building a DNN-based detection framework is the main emphasis of this study. Some 450,000 new malware programs and possibly undesirable apps aimed at specific mobile devices are released every day, according to a research. Banking trojans, spyware, adware droppers, and other forms of malware are increasingly targeting cellphones, according to security experts. Of particular note is the fact that 3.55 million different kinds of Android applications can be found in only the Google Play Store. Still, Android users have the option to install software from unofficial sources, which opens the device up to the possibility of downloading harmful programs from untrustworthy servers.

In order to create an Android app, developers need to add certain files in the package format that ends in.apk. Important app information may be found in AndroidManifest.xml, which includes package version, necessary permissions, intents, actions, and services. The classes.dex file contains all of the bytecode that specifies how the program works. Android uses a permission-based security approach to improve the security framework by allowing applications only the rights that the user explicitly grants. Android continues to face continued issues in protecting itself from developing cybersecurity threats, since these security measures have not been able to remove the varied spectrum of assaults.

The majority of antivirus programs use signature-based detection, which is easy for malware developers to circumvent by changing the signature of the harmful program. Malware code can also avoid detection by using little obfuscation. Users must be able to detect Android malware, and many researchers have suggested different ways to lessen the impact of these assaults, showing that the platform is being constantly fortified against harmful threats.

Incorporating a diversified variety of features during training may enhance detection models' efficacy; hence, it is important to explore a wide range of characteristics for robust model development. Although there are various detection frameworks available, this study's literature review shows that most of them train on a small set of characteristics in order to keep model complexity down. Because of this restriction, it may be more difficult to identify malicious apps

in a dataset if their characteristics are excluded from the feature selection procedure. The detection model could be prone to mistakes if such features are ignored. Alternatively, it is possible to enhance the probability of identifying malicious programs by taking into account a large number of characteristics as input for training a model.

II. REVIEW OF LITERATURE

Almarshad, F A et al., (2023) In recent years, there has been a trend toward Android malware protection solutions that can quickly detect and classify different types of malware in order to develop strategies for rapid reaction. A lack of data for malware samples has been mentioned as a problem with developing effective deep learning-based solutions, even though many application sectors have shown the benefits of using these approaches to automate and deliver self-learning services. To get over this problem, this research proposes a Siamese neural network that is based on one-shot learning; it can detect malware attacks and classify them into different groups. Our proposed approach makes use of the Drebin dataset, which classifies components as either harmless or dangerous. The database dataset consists of 5,560 Android malware apps and 9,476 goodware apps, which are used to evaluate the efficacy of the recommended technique. The five most important steps in putting it into action are preparation, data segmentation, model design, training, and evaluation. Using N-way one-shot tasks, the accuracy is measured in both the training and testing stages. Siamese networks are trained to rank sample similarity. With a 98.9% success rate, our Siamese Shot model outperformed the industry norms in this trial. Furthermore, Keras and TensorFlow are the most popular frameworks.

Vipin Kumar and Shyam Dwevedi. (2022) Due to the digitization of several services and the inexpensive cost of smartphones, their adoption has skyrocketed in the past few years. The major cause for concern in this study is the proliferation of malware, which has arisen as a result of the proliferation of smartphone use. The exponential growth of malicious mobile apps, especially on the Android platform, has made it nearly impossible to detect potentially dangerous applications. As the number of Android devices continues to rise, virus developers continue to release new malicious software that might compromise the security of the system and the personal information of its users. Using machine learning techniques to identify Android malware is the focus of this thesis. We provide an Android malware detection system that uses six different machine learning

models—Decision Trees, Support Vector Machines, Naive Bayes, Random Forests, K-Nearest Neighbors, and Ensemble Methods / Extra-Tree Classifier—to classify different types of malware. The CICMalAnal2017 Android malware dataset is used to evaluate the performance of the suggested framework. Adware, ransomware, and scareware are among the malicious programs found in the CICMalAnal2017 dataset. There are four different kinds of feature selection procedures that are utilized: information gain, chi-square test, random forest importance, and feature correlation. The categorization of malware is an area where several machine learning algorithms are being evaluated. Given the potential difficulty of implementing security measures after the program has been deployed, ML-based methods for evaluating source code vulnerabilities are eventually detailed. Therefore, this study aims to shed light on the subject and point academics in the direction of potential future research and development avenues.

Sk, Khader Basha. (2022) When it comes to software and operating systems, malware is a big problem. Not only that, but the Android system is experiencing the same issues. There have been previous examples of malware detection methods that relied on signatures. Still, the methods failed to identify any previously unseen virus. An important problem remains, even if there are many methods for detection and analysis, and that is the precision with which new viruses can be detected. Existing approaches for detecting and analyzing Android malicious code are studied and highlighted in this research. We will be undertaking semantic analysis in addition to our studies, and we have proposed machine learning techniques to assess this type of malware. Potentially harmful apps will have access to a database of permissions. Those will be contrasted with the application permissions that we intend to examine. By the end, not only can we assess the program using comments, but the user will also be able to view the amount of dangerous permission it has.

Pandey, Sonal et al., (2021) Mobile malware has also increased dramatically, thanks to Android's open architecture. Malicious apps continue to evolve in quantity, variety, complexity, and variation, making it difficult for traditional approaches to identify them. Despite their efficacy against known malware, signature-based solutions are unable to identify novel or undiscovered forms of malware. Using machine learning techniques, the author of this research will implement a method to identify new Android viruses. Our method achieves great accuracy by extracting characteristics related to permissions (both AOSP and third party permissions). After that, features were chosen for the training and testing classifiers in tandem with their respective apks, which

included both malicious and benign files. The AndroZoo dataset, which contains 15,000 malicious and 15,000 benign Apks, is used to test our technique. Our results using AOSP and Third Party Permission show that our Random forest classifiers can accurately categorize Android malware with 91.1% and 72.3% accuracy, respectively.

Yildiz, Oktay & Doğru, İbrahim. (2019) Android, an open-source mobile operating system (OS) based on Linux, has surpassed all others in popularity as the number of smartphones continues to rise. Because Android is so popular, most spyware focuses on Android devices and consumers. It is for this reason that the development of malware detection solutions for Android smartphones is crucial. Analysis and detection of Android malware are seeing a rise in the usage of machine learning approaches. A feature selection using genetic algorithm (GA) technique for identifying Android malware is presented in this paper. Using GA, we chose three distinct classifier algorithms with various feature subsets to identify and analyze Android malware in a comparative manner. With 16 chosen permissions and a dataset of 1740 samples (1,191 malware samples and 621 benign samples), the greatest accuracy result was 98.45% achieved by combining Support Vector Machines with a GA.

Tchakounte, Franklin. (2014) A permission-based system that limits access to important resources on an Android device by third-party apps has been introduced into Android security. Before an app can be installed, the user must confirm that they are okay with the permissions it requests. The goal of this procedure is to warn users about potential dangers before they install and use an app on their device. However, even when the permission system is clear, users still don't know enough about the danger to trust the app store or the app's popularity to question the developer's motives and install the app anyway. Methods to classify malware using permissions, either separately or in combination, using machine learning classifiers are being developed at a rapid pace. Based on the above, this work aims to research existing methods for malware characterization and detection in the literature. In doing so, we highlight both the shortcomings of prior investigations and the encouraging aspects of potential future studies.

III. RESEARCH METHODOLOGY

Data Collection Phase

In order to get this data, we needed datasets in.apk files from both safe and malicious apps. This part involves selecting samples at random from AndroZoo databases. In order to provide its analysis, AndroZoo gathered an Android application executable from several sources and made it public. Moreover, the dataset only included apps that were retrieved from the Google Play store. Also, malicious apps that endanger consumers may be removed by the Bouncer detector that Google Play has installed. Consequently, it is more correct to get apps from Google Play, since they usually generate applications for the benign applications dataset.

Decompiling APK File

The AndroidManifest.xml file is a key resource for gathering data about an app, including its permissions and actions. The extracted permissions should be saved as an x.arff file and then imported into WEKA. The values of the permissions are kept as binary numbers, either 0 or 1. Gaining the greatest benefits of permissions is also made easier with the optimization option. To train and categorize the malware detection permission characteristics, major features were used. An excellent feature for the most effective malware detection was obtained in this research by using the features selection method. To ensure that the prediction model is as accurate as possible, features selection techniques sift through attribute data and exclude those that are irrelevant or inappropriate. This resulted in a decrease from 20 to 15 malware characteristics for each authorization level. In order to distinguish between safe and malicious content, this is necessary. Multiple iterations of tenfold cross-validation were also used in this investigation.

Machine Learning Classifier

One kind of AI, known as machine learning, can pick up new skills automatically, without any human intervention. When exposed to fresh information, it may foretell the future and enhance decision-making. The process of making predictions is often known as learning, and it is based on searching through data sets for patterns. Different kinds of classifiers provide different learning processes and prediction outcomes. In the field of intrusion detection systems, this method is often used for sample classification, especially in the benign and malware domains. The study has used

a supervised machine learning technique due to the presence of labels (malware and benign) in the sample data set.

In addition, by reducing errors, supervised machine learning provided excellent results. Various kinds of machine learning classifiers have been used to interpret each of the five classifiers throughout the study. The Random Forest (RF), J48, Multi-Layer Perceptron (MLP), DecisionTable, and Naïve Bayes are the five classifiers.

The research also examined the various metrics of each classifier using factors such as accuracy, FPR, precision, recall, and f-measure.

IV. RESULTS AND DISCUSSION

Table 1: Performance results of classifiers

Classifiers	Accuracy (%)	FPR	Precision	Recall	F-measure
Random Forest	89.40	10.68	89.38	89.38	89.38
Naïve Bayes	85.98	14.02	86.11	86.11	86.11
J48	89.21	10.79	89.17	89.17	89.17
Decision Table	89.15	10.83	88.98	88.98	88.98
MLP	89.09	10.85	88.98	88.98	88.98

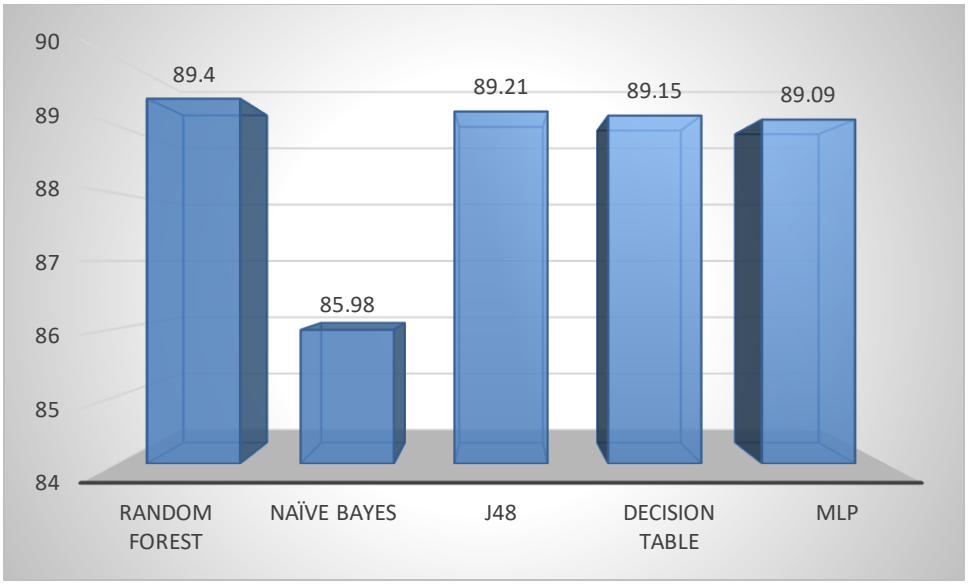

Figure 1: Accuracy of classifiers

A total of 89.40% accuracy and a False Positive Rate (FPR) of 10.68% were attained using Random Forest, according to the performance assessment of the five classifiers. The fact that it has F-measure values of 89.38% for accuracy, recall, and balance adds credence to its consistent and balanced predictive power. With an accuracy of 89.21% and an FPR of 10.79%, J48 has similar performance but somewhat worse consistency across precision, recall, and F-measure. With accuracies of 89.15% and 89.09%, respectively, coupled with precision, recall, and F-measure values around 88.98%, the Decision Table and MLP classifiers provide modest but consistent performance, suggesting their dependability but somewhat diminished efficacy when contrasted with Random Forest. Although it's computationally efficient, Naïve Bayes has the lowest accuracy at 85.98% and the greatest false positive rate of 14.02%, which shows that it can't handle complicated patterns in the data very well.

Table 2: Confusion Matrix of classifiers

Classifiers	**Actual**	**Prediction**	
		Benign	**Malware**
Random Forest	Benign	9065	935

	Malware	1398	8602
Naïve Bayes	Benign	9227	773
	Malware	1989	8011
J48	Benign	9125	875
	Malware	1178	8822
Decision Table	Benign	9146	854
	Malware	1500	8500
MLP	Benign	9150	850
	Malware	1356	8644

The findings of the confusion matrix show that there are significant variations in the way each classifier differentiates between safe and dangerous applications. Although it incorrectly identified 935 benign instances as malware and 1398 malware instances as benign, Random Forest demonstrates good discriminative capabilities by properly recognizing 9065 benign and 8602 malicious samples. J48 outperforms Random Forest in terms of accuracy in detecting malware, with 9125 true benign and 8822 genuine malware predictions. It also has a lower number of false negatives (1178) compared to Random Forest. The MLP classifier shows consistent results as well, successfully identifying 9150 benign and 8644 malware occurrences; nevertheless, it still has greater false negatives (1356) compared to J48. However, the Decision Table classifier has a greater number of missed malware instances (1500), which makes it less reliable for security-sensitive applications. It accurately labels 9146 benign and 8500 malicious samples, which is intermediate. Although Naïve Bayes is computationally fast and simple, it has the greatest misclassification rates, especially when it comes to malware detection, with 1989 samples wrongly identified as benign. This may lead to serious dangers.

V. CONCLUSION

In this research, we found that a combination of permission-based characteristics and supervised machine learning algorithms is the best way to identify malware on Android. Researchers were able to determine which machine learning models were the most effective by extracting and optimizing important permissions from APK files and then comparing several classifiers using tenfold cross-validation. Among the classifiers, Random Forest achieved the best accuracy and the lowest false positive rate, while J48 and MLP also demonstrated dependable classification skills. The reason Naïve Bayes was limited was that it misclassified malware samples more often. A tree-based algorithm's ability to capture complicated permission patterns and differentiate between good and bad apps was further validated by the confusion matrix analysis.

REFERENCES: -

[1] F. A. Almarshad, M. Zakariah, G. Gashgari, E. Aldakheel, and A. Alzahrani, "Detection of Android malware using machine learning and Siamese shot learning technique for security," *IEEE Access*, vol. 10, pp. 1–10, 2023.

[2] N. Chrysikos, P. Karampelas, and K. Xylogiannopoulos, "Permission-based classification of Android malware applications using Random Forest," in *Proc. Eur. Conf. Cyber Warfare Secur.*, vol. 22, pp. 132–142, 2023.

[3] H. AlOmari, Q. Yaseen, and M. Al-Betar, "A comparative analysis of machine learning algorithms for Android malware detection," *Procedia Computer Science*, vol. 220, pp. 763–768, 2023.

[4] M. Ibrahim, A. Abdullahi, M. A. Ahmad, R. Mustapha, and M. Ng, "A comparative analysis of Android malware detection with and without feature selection techniques using machine learning," *SLU Journal of Science and Technology*, vol. 6, no. 1–2, pp. 235–246, 2023.

[5] H. Negi, "Android malware detection using machine learning," *Int. J. Recent Technol. Eng.*, vol. 8, no. 2, pp. 65–70, 2022.

[6] V. Kumar and S. Dwevedi, "Android malware detection using machine learning techniques: A review," *Dogo Rangsang Res. J.*, vol. 9, no. 1, pp. 109–117, 2022.

[7] K. B. Sk, "Detection of malware in Android using machine learning," *Mukt Shabd J.*, vol. 11, no. 6, pp. 62–67, 2022.

[8] F. Akbar et al., "Permissions-based detection of Android malware using machine learning," *Symmetry*, vol. 14, no. 4, pp. 1–19, 2022.

[9] S. Pandey, Satyasheel, and D. Jain, "Permission-based Android malware detection using machine learning," *Int. J. Fundamental Appl. Sci.*, vol. 10, no. 1, pp. 16–20, 2021.

[10] N. Nasri, "Android malware detection system using machine learning," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 1, pp. 327–333, 2020.

[11] O. Yildiz and İ. Doğru, "Permission-based Android malware detection system using feature selection with genetic algorithm," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 29, no. 2, pp. 245–262, 2019.

[12] R. S. Arslan and İ. Doğru, "Permission-based malware detection system for Android using machine learning techniques," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 29, no. 1, pp. 43–61, 2019.

[13] F. Tchakounte, "Permission-based malware detection mechanisms on Android: Analysis and perspectives," *J. Comput. Sci. Softw. Appl.*, vol. 1, no. 2, pp. 63–77, 2014.

[14] Z. Aung and W. Zaw, "Permission-based Android malware detection," *Int. J. Sci. Technol. Res.*, vol. 2, no. 3, pp. 228–234, 2013.