iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** VII **Month of publication:** July 2022

DOI: https://doi.org/10.22214/ijraset.2022.45726

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VII July 2022- Available at www.ijraset.com

Detection of Malware in Android Phones Using Machine Learning

Poojitha K[1], Dr. Usha J[2]
[1, 2]*RV College of Engineering, RV Vidyanikethan Post, Mysuru Road, Bengaluru 560059, Karnataka*

Abstract: ***In a major cyber security scare, around 1.5 crore Android devices in India have been infected by malware without the knowledge of the users. According to a report by cyber security solution firm Check Point Research in 2020, a new variant of mobile malware has quietly infected around 2.5 crore devices worldwide. Malware is any type of malicious software or code designed to harm a user's device, such as trojans, adware, ransomware, spyware, viruses or phishing apps. The permissions and API-calls are extracted from all Android applications, and both were included as features in the dataset. It is a process of analysing the malware binary without running the code. To offer a simple, streamlined, document-centric experience Jupyter Notebook interactive development environment and Flask is utilized. The Androguard tool and genetic algorithm analyses the APK files by separately extracting the permissions for each APK file. Supervised Machine Learning algorithms used are Support Vector Machine (SVM) and MLP an ANN numeral network approach is used to compare the traditional machine learning techniques. Experiments will be conducted on two types of models, traditional machine learning classifiers and deep learning neural networks. Initially, the classifiers are trained using the dataset, taken from android malware dataset and then testing and evaluation is performed based on the extracted features. An efficient method to detect the presence of malware in the android mobile phones using the permissions, API calls is implemented and the best classifier is identified which gives optimal results with accuracy, F-measure, Recall, and Precision scores. This would enable users to easily navigate various resources available with an adaptive user interface using android application.***
Keywords: Malware classification, SVM,MLP

I. INTRODUCTION

Malware includes any forms of harmful software and code intended to damage a user's device, including trojan horses, adware, ransomware, spyware, viruses, and phishing software. In this project, we go over the creation of machine learning (ML) models for malware detection. The machine learning model creates a framework for a web application where we can analyse if apk files are malicious or benign files and classify them using ML classifiers utilising datasets on Android malware.

The suggested methodology includes two interpretability- different ML models. The ability to extract permissions and API calls from all Android applications and incorporate them as features in the dataset demonstrates how Android malware data and ML algorithms can be used for the reasonably priced prediction of harmful files.

The solution that is being proposed uses machine learning to anticipate malware analysis. Only the malware data from Android that has been uploaded has been used for the analysis. The data are pre-processed using a general algorithm and the androguard tool to extract the features, and they are trained so that the system can anticipate the analysis of the entered data and provide a graphical representation through graphs. The suggested method is a web application framework that uses Flask for front end technologies, HTML, CSS, JavaScript, and for validations. The main side language in this project is Python, an object-oriented programming language.

II. LITERATURE SURVEY

For the purpose of identifying malicious Android applications, machine learning techniques have been deployed. data pre-processing, which is used for data features. Techniques for feature extraction such data trimming are applied. Malware analysis uses the RF (Random Forest) technique, LR (Logistic Regression), NB (Naive Bayes), SVM (Support Vector Machines), and K-NN (K-nearest neighbours). Multiple ML algorithms provide precise implementation, testing, and training. For each individual method, the procedure of detecting malware has been constrained. Each classifier has been demonstrated by utilising the ROC curve to indicate which algorithms are more effective than others. While researching and evaluating classifiers, the SVM plus Random Forest combination performed decently in comparison to other algorithms. [1]

The Android device used as an attack point the most frequently by attackers is the subject of a study on the detection of malware. 32 resource features are tracked in this work with the goal of identifying Android malware. Among feature-selection algorithms, Information Gain is chosen. Additionally, performance evaluations of machine learning classifiers for malware detection are conducted. Data is gathered from Android-powered smartphones in order to identify malware in an Android environment. In order to assess how well machine learning classifiers performed on the datasets that were gathered, this article used 10-fold cross-validation. The experimental findings demonstrated that the Random Forest classifier's performance was best in terms of TPR/FPR. [2]

To identify fraudulent Android Apps, permissions and API (Application Program Interface) calls are merged, and machine learning techniques are used. The goal is to establish that increasing accuracy may indeed be gained by combining API calls with permissions. A highly thorough coverage of Android malware is indicated by the 610 malware samples spread across 49 different malware families.

The classification techniques used to select the best classifier include SVM, Decision Trees, and bagging. The findings demonstrate that the Bagging predictor has the highest detection rate in each example, whereas the decision tree technique (J48) has the lowest detection rate. When applied to data with skewed class distributions, bagging performs well. [3]

It has been suggested to employ a very original humanoid malware detection framework that uses choices like permissions to duplicate the behaviour of the application. From distinct apk files, completely different kinds of permissions are retrieved. EDA, or exploratory data analysis, is used to locate the missing values. While compared to categorical data, superior results were found when using random forest for feature selection. Androguard, an APK analyzer that only functions on Apk files, was used to test these programmes. The malicious programmes are examined and downloaded from virus-share, a collection of malware samples. The ultimate accuracy and prediction are affected by the optimizer like Adam, SGD, and activation like Relu, Tanh, and sigmoid.The complete model has been put through testing using a variety of parameters, optimizers, feature techniques, and other layers for the model's improvement. According to system testing results, utilising a deep learning model to identify malware has demonstrated to reach high accuracy of 94.65 percent. [4]

Malware behaviour in the Android platform is being researched and analysed, using both static and dynamic malware analysis techniques. The AndroidManifest.xml's features are extracted for the static analysis. To extract data from APK forms, utilise the AndroGuard programme.

Static analysis has made use of the classifier methods Support Vector Machine (SVM), Logistic Regression (LR), and K-Nearest Neighbor (KNN). For the investigation of harmful programmes in real time, use Droidbox. Support Vector Machine, K-Nearest-Neighbor Classifier, Decision Tree, Logistic Regression, and Random Forest Classifier are some examples of the supervised machine learning models that were tested in the dynamic study. The best F1-score in dynamic analysis was provided by the Random Forest classifier. Static analysis has shown a better accuracy rate of 81.03 percent using logistic regression. The accuracy scores of the dynamic analysis have reached over 93 percent, significantly surpassing those of the static analysis. [5]

III. SUMMARY OF LITERATURE REVIEW

The main motivation behind this project is that it has been identified that users are responsible for most security issues. At the time of installing Android applications, users will be asked to allow some permissions. However, all the users may not understand the purpose of each permission. They allow permission to run the application without considering the severity of it. Fraudulent applications might steal data and perform unintended tasks after getting the required permissions

A. Existing System

For malware analysis, the current system utilised and processed one of the most trustworthy datasets ever generated. With the help of the Androguard programme and a general algorithm, a collection of features pertaining to the Android virus are extracted from this information. All Android applications' permissions and API calls are gathered and included as features
in the dataset. In order to achieve this, two algorithm classifiers are described, one of which, the Multilayer Perceptron (MPL), was chosen due to its high accuracy. In spite of SVM's poorer
accuracy compared to other models, it was chosen over the others because of its explainability. Utilizing permissions and API calls, a reliable technique is put into place to find malware on Android mobile devices, and the best classifier is found to produce the greatest results in terms of accuracy, F-measure, Recall, and Precision scores. This would make it possible for users to quickly navigate through the numerous resources that are available with an Android application's adaptive user interface.

B. Proposed System

The dataset is drawn from the android malware dataset, which combines 345 benign samples with 365 malware samples, with the purpose of detecting android malware. The permissions and API-calls characteristics of the Android application are extracted using the Andropy tool and a general method. The utility incorporates the retrieved features and prepares them for storage in csv files. On the basis of the classification report, the SVM and MLP machine learning algorithm classifiers are used to assess the accuracy for identifying malware and to find the best classifier with the highest accuracy.

IV. SCOPE OF THE PROJECT

1) to gather the dataset, which includes both malicious and clean apk files.
2) Utilizing the general technique, extract the AndroPy tool's common characteristics from the apk files.
3) Pre-process the features and store them in a csv file To create the SVM machine learning algorithms for categorisation and assessing the outcomes
4) Building a Multilayer Perceptron (MLP) ANN neural network for identifying malicious or benign apk files

V. ARCHITECTURE

Figure 1. The system overview shows the numerous procedures that are taken in order to find malware in Android applications. The block diagram consists of the steps in takes the dataset which in apk format and the data is pre-processed and features are extracted from the andropy tool, the data is then sent to the Ml classifiers the detect the presence of malware in the apk files and also displays the classification report.

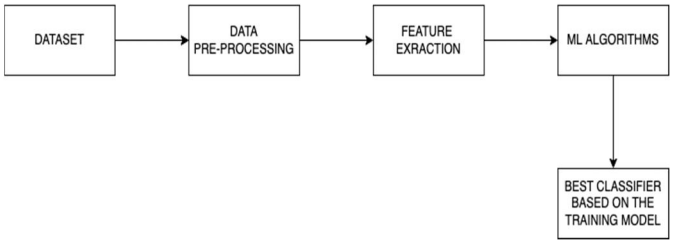


Fig. 1: Block Diagram of the System

VI. METHODOLOGY

The suggested methodology includes two interpretability- different ML models. This demonstrates that the malware dataset for Android and ML classifier can be used to predict malware in certain apk files cost-effectively because the dataset was acquired with both malicious and benign files. Computers may now learn without explicit programming thanks to the branch of study known as machine learning. One of the most intriguing technologies one has ever encountered is machine learning (ML). As implied by the name, it gives the computer characteristics that make it more like people. The problem definition is categorised using machine learning algorithms in order to find the best solution and accomplish the objectives.

VII. RESULT AND DISCUSSION

The experiment shows that multilayer perceptron (MLP), an ANN neural network, has the highest accuracy of 92.26 percent in comparison to SVM, making it the top classifier. Based on unit testing, the module specification is created. Additionally, test cases enable the user to construct the project in a way that makes it simple for them to comprehend the kinds of apk files that should be used for classification based on their features, for malware detection, and for determining if a file is malicious or not.The results also include a classification report with the number of samples for each sort of malware feature, including benign, and provides precision, recall, and f1-score percentages. The proportion of genuine positives to the total of both true and false positives is known as precision. Recall measures the proportion of genuine positives to the total of both true positives and false negatives. The hormonic mean of precision and recall is the F1-score. Support is the volume of malicious or benign files of a given type.Fig 2 displays the malware detection for the calendar.apk file. We assume that the classification is carried out using the MLP neural network, which takes into account the features in the apk file and outputs a benign file with a model accuracy of
92.26 percent. Similar to this, we can choose the type of algorithm for various apk bugs and determine the accuracy percentage to identify the file contaminated with malware or whether it is a safe file for Android mobile devices to use..

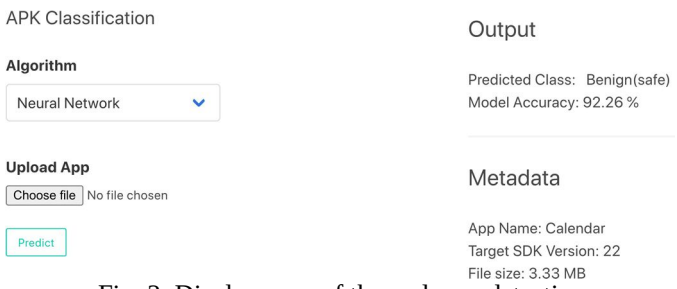


Fig. 2: Display page of the malware detection

VIII. CONCLUSION

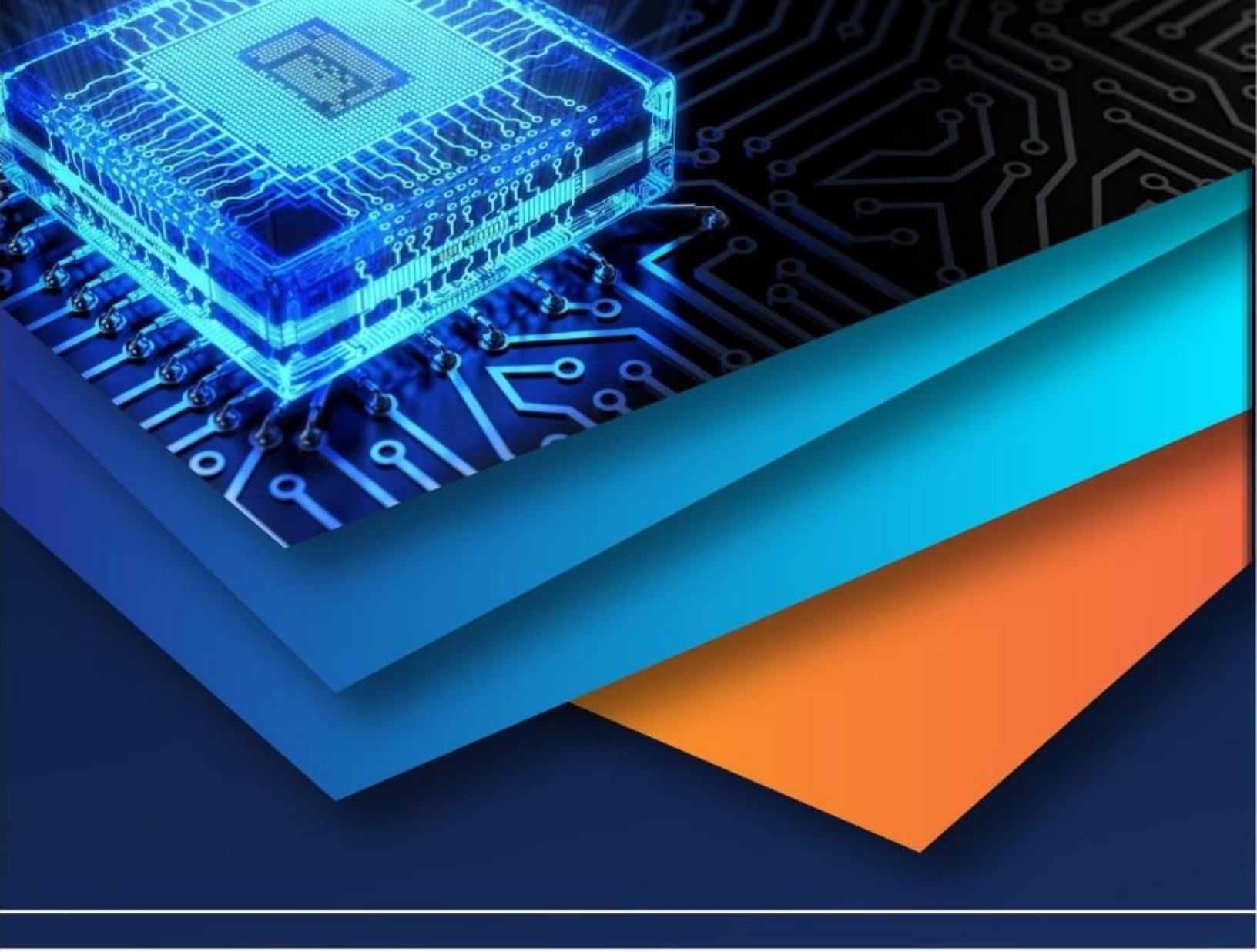
Malware has quickly developed into a serious security risk for the computing industry, which makes it one of the main causes of the majority of the current security issues on the Internet. Malicious code continues to be a significant hazard on the Internet in spite of the fact that a lot of study has been put into malware detection. Recently, numerous Malware detection methods and procedures have been suggested to address these issues. These methods and strategies, however, have some drawbacks that prevent them from solving the issue. Malware attacks are on the rise due to the increased use of mobile devices, particularly on android smartphones, which hold 72.2% of the market. It also shows that most security problems are caused by users. Users will be prompted to grant permissions while installing an Android application. However, it's possible that not every user is aware of each permission's function. They permit them to use the application without taking its harshness into account. After obtaining the necessary permissions, malicious applications may steal the data and carry out undesired functions.

IX. ACKNOWLEDGMET

We express our sincere thanks and wholehearted credit to our internal guide Dr. Usha J, Professor, Department of MCA, R.V. College of Engineering ®, Bengaluru for his constant encouragement, support and guidance during the Project work.

REFERENCES

[1] S. Priyadharshini and S. Shanthi, "A Survey On Detecting Android Malware Using Machine Learning Technique," 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), 2021, pp. 1621-1627, doi:10.1109/ ICACCS51430.2021.9441712.
[2] Hyo-Sik Ham and Mi-Jung Choi, "Analysis of Android malware detection performance using machine learning classifiers," 2013 International Conference on ICT Convergence (ICTC), 2013, pp. 490-495, doi: 10.1109/ICTC. 2013.6675404.
[3] N. Peiravian and X. Zhu, "Machine Learning for Android Malware Detection Using Permission and API Calls," 2013 IEEE 25th International Conference on Tools with Artificial Intelligence, 2013, pp. 300-305, doi: 10.1109/ICTAI.2013.53.
[4] S. HR, "Static Analysis of Android Malware Detection using Deep Learning," 2019 International Conference on Intelligent Computing and Control Systems (ICCS), 2019, pp. 841-845, doi: 10.1109/ICCS45141.2019.9065765.
[5] U. S. Jannat, S. M. Hasnayeen, M. K. Bashar Shuhan and
[6] M. S. Ferdous, "Analysis and Detection of Malware in Android Applications Using Machine Learning," 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), 2019, pp. 1-7, doi: 10.1109/ECACE. 2019.8679493
[7] N. Cristianini and J. Shawe-Taylor, An introduction to Support Vector Machines and
[8] other kernel-based learning methods. Cambridge University Press, March 2000.
[9] B. Sch¨olkopf and A. J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond. MIT Press, 2002.
[10] P. Baldi, S. Brunak, Y. Chauvin, C. A. Andersen, and H. Nielsen,"Assessing the accuracy of prediction algorithms for classification," Bioinformatics, no. 5, pp. 412–424, May 2000.
[11] S. N. N. Kwang Loong and S. K. K. Mishra, "De novo SVM classification of precursor microRNAs from genomic pseudo hairpins using global and intrinsic folding measures." Bioinformatics, January 2007.
[12] Y. Freund and R. E. Schapire, "Large margin classification using theperceptron algorithm," in Machine Learning, vol. 37, 1999, pp. 277–296.
[13] C. Cortes and V. Vapnik, "Support-vector networks," in Machine Learning, 1995, pp. 273–297.
[14] V. N. Vapnik, The Nature of Statistical Learning Theory (Information Science and Statistics). Springer, November 1999
[15] M. R. Chouchane, A. Walenstein, and A. Lakhotia, "Using MarkovChains to filter machine-morphed variants of malicious programs,"in Malicious and Unwanted Software, 2008. MALWARE 2008. 3rd International Conference on, 2008, pp. 77–84.
[16] M. Stamp, S. Attaluri, and S. McGhee, "Profile hidden markov modelsand metamorphic virus detection," Journal in Computer Virology, 2008.
[17] R. Santamarta, "Generic detection and classification of polymorphicmalware using neural pattern recognition," 2006.
[18] I. Yoo, "Visualizing Windows executable viruses using self-organizingmaps," in VizSEC/ DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security. New York, NY, USA: ACM, 2004, pp. 82–89.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY