International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 05 | May - 2023 Impact Factor: 8.176 ISSN: 2582-3930

ANDROID MALWARE DETECTION USING MACHINE LEARNING

Mrs.S..Francis Shamili, M.E.,
Department of Computer Science and Engineering,
Dhanalakshmi Srinivasan Engineering College, Perambalur.

Anumala Venkatesh
Department of Computer Science and Engineering
Dhanalakshmi Srinivasan Engineering College, Perambalur.

Gavini Ganesh
Department of Computer Science and Engineering
Dhanalakshmi Srinivasan Engineering College, Perambalur.

Moraboina Thirupati Raju
Department of Computer Science and Engineering
Dhanalakshmi Srinivasan Engineering College, Perambalur.

Muram Thirumaleshwar reddy
Department of Computer Science and Engineering
Dhanalakshmi Srinivasan Engineering College, Perambalur.

Reddimi Manoj Kumar Reddy
Department of Computer Science and Engineering
Dhanalakshmi Srinivasan Engineering College, Perambalur.

***Abstract*— One of the most popular operating systems for mobile devices is Android. Similar to the growth of mobile devices, harmful apps and adware are becoming more and more prevalent. There are several commercial technologies based on signatures that are on the market that may limit the penetration and spread of malicious programmes. According to multiple studies, standard signature-based detection systems are effective up to a point and malware writers use a variety of evasion tactics to get around these tools. Therefore, given the current situation, a replacement for the signature-based system that is very effective at detecting malware is becoming more and more necessary. Machine learning techniques that assess data from harmful apps and utilize those attributes to identify and detect unknown dangerous programmes were the subject of much recent study. In this model, we introduced a brand-new malware detection method based on Android permissions. The use of machine learning methods, such as the Logistic Regression Model, Random Forest, Support Vector Machine and Neural Network, is our last step.**

I. INTRODUCTION

In today's society, cellphones are essential to daily life and communication. In 2022, there were 3.8 billion smartphone users worldwide, and in the years to come, that figure is expected to rise by several hundred million. Over 96% of smartphones are powered by Android or iOS, and Android smartphones account for about 54% of all mobile traffic worldwide. Despite the influence of technologies, new attacks that have the potential to compromise the security of smartphones are becoming less frequent every day. The most common operating system is Android, and

Android handsets are well-liked by attackers who are aiming to compromise sensitive data kept on smartphones. Hackers are distributing malware using Android and Google Play Store apps.

There is a possibility that the amount of danger posed by the virus varies; it may be something as innocuous as an advertisement that appears every few minutes, or it could be something as severe as accessing a victim's bank account. No matter what happens, everything is going to rely on how the infection behaves. Crypto miners degrade device performance and deplete the battery by using the device's hardware resources to mine bitcoin for the malicious actor who installed them. Adware is the cause of the invasive advertising. Ransomware is a distinct threat that may encrypt the contents of a device until the attacker receives a ransom payment. Malware that gains root access to the device gives the attacker access to personal information that may be exploited to access the victim's accounts. The attacker is then able to use the phone as if it were their own, browse it, and do other activities.

II. RELATED WOR

A summary of earlier work on android malware detection is provided. A unique Android malware detection framework was suggested in 2018 by TaeGuen Kim, BooJoong Kang, Mina Rho, Sakir Sezer, and Eul Gyu Im[1] that makes use of several static attributes to reflect the characteristics of apps in multiple ways. In the end, they used the multimodal deep learning approach, which is designed to handle a variety of feature types.

In 2019, Jiaqi Pang and Jiali Bian[3] introduced a strategy for Android malware detection that uses a static analysis approach to extract three pieces of information: requested permissions, system API calls, and the ap. By changing the ap threshold, we get better detection results. We also provide a

quick and effective way for detecting several unidentified Android applications.

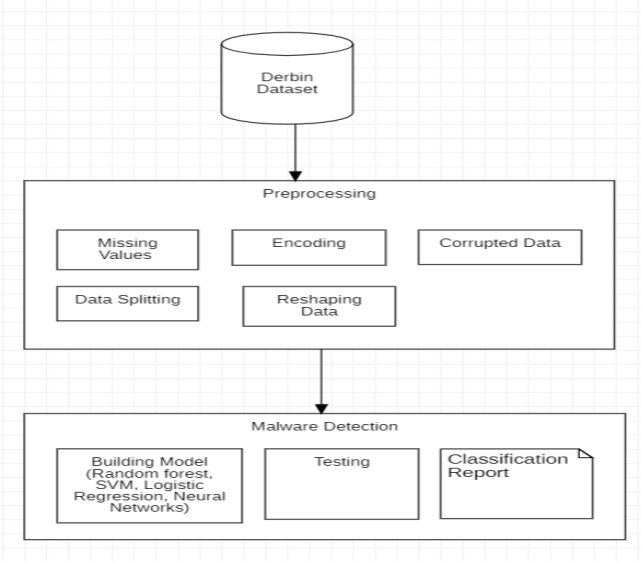
In the year 2020, Ahmed Hashem El Fiky, Ayman Elshenawy, and Mohamed Ashraf Madkour [9] announced a universal machine learning classifier that was based on multi-level feature selection approaches. In order to develop a binary-level machine learning classifier, PCA and IG techniques were used. For the chosen dataset, the developed classifier results in an 89% decrease in the number of initial features. The testing findings demonstrated that, when compared to other algorithms, the RF classifier had the best performance for identifying Android malware applications.

In 2021, Huijuan Zhu, Yang Li, Ruidong Li, Jianqiang Li, Zhuhong You, and Houbing Song[12] developed a system with a two-tier architecture that combines base learner output from SVM with an ensemble of MLP base learners. The variety of the training subsets is ensured at the initial step by the twofold disruption of feature space and sample space, and PCA is performed on these subsets independently. To ensure the correctness of the base learner, MLP is applied to each branch, maintaining all principle components obtained by the PCA and changing the whole training dataset into a completely new set.

Hyun-Il Kim, Moonyoung Kang, Seong-Je Cho, and Sang-Il Choi[13] will all be in 2022. Our technique extracts input data in the form of strings from every part of a malicious app's certificate file (CR), executable code (CL), and manifest file (AM), respectively. The data are then connected in a series, and a 1D convolution filter-based deep learning model is created. This model gives it the ability to eventually categorise harmful programmes into real-world malware families. 5530 VOLUME 10, 2022 H.-I. Kim et al.

III. METHODOLOGY

ARCHITECTURE DIAGRAM:



1) DATA EXPLORATION:
Viewing and presenting data is the initial phase in data analysis, and it is done in order to get early insights or identify areas and trends that need more research. Point-and-click data exploration and interactive dashboards may help users gain insights more quickly and see the broader picture more clearly. Before posing more detailed questions, users may make wiser decisions on where to go further into the information and get a full understanding of the business. using data investigation as a first step. A user-friendly interface may assist everyone in an organization become familiar with the data, spot trends, and ask insightful questions that could lead to a more complete and productive research.

2) DATA CLEANING:
Data cleaning is the process of examining a record set, table, or database to identify and correct incorrect or inaccurate data. It is vital to identify the portions of the data that are incorrect, incomplete, or irrelevant, and the erroneous or inaccurate data must then be corrected, amended, or eliminated. The data must then be transferred. When erroneous data is used, despite the seeming correctness of the results and algorithms, they are erroneous. Since procedures vary There is no one approach that can be used universally to identify the precise actions that make up the data cleaning process. This is because each dataset is unique. It is very necessary to design a pattern for your data cleaning process so that you can reliably go through each step in the right order.

3) DATA VISUALISATION:
Data visualization is an integral part of the data science process, helping teams and individuals convey insights to their colleagues and superiors. Management teams in charge of reporting systems usually employ standard, pre-made views for this purpose. However, data visualization is not analyst zero in on relevant themes, patterns, and hidden relationships in the otherwise unstructured data. Furthermore, they might be used to illustrate the relationships between nodes in a graph-based knowledge network.

4) DATA PREPARATION:
One of the most challenging aspects of any machine learning endeavour is undoubtedly getting the data ready for analysis. This is due to the fact that each dataset is one-of-a-kind and was developed for specific needs. Despite this, we can identify a common sequence of the jobs and subtasks you will likely complete across the various predictive modelling projects. This procedure provides a structure within which we may evaluate the data preparation needed for the project, factoring in the project description completed before to the data preparation and the algorithm evaluation performed after the data preparation.

5) MODEL CREATION:
Machine learning models often need huge quantities of high-quality training data to be accurate. The connection between the input and output data will be taught to the model using

this training dataset. Depending on the kind of machine learning training being done, these datasets' make-up will change. Labeled datasets with input and output variables that have been labeled are used to train supervised machine learning models.

A) RANDOM FOREST :

The random forest is a method of categorization that uses several decision trees. Each random forest tree makes a forecast for a class, and the class with the most votes determines the model's prediction. For categorization problems, a random forest returns the most popular classification as its verdict. When doing a regression, the average prediction from each tree is given as the result. When compared to decision trees, random choice forests mitigate the latter's tendency to overfit the data in their training set. When compared to random forests, decision trees generally fall short, however gradient enhanced trees do far better. The critical importance of the low correlation between models. It's possible that an ensemble of projections from uncorrelated models might be more accurate than any one model's forecast taken individually. Trees have this remarkable impact because they watch out for one another and correct one other when they make mistakes.

B) SUPPORT VECTOR MACHINE :

The Support Vector Machine, often known as an SVM, is one of the most well-known algorithms for supervised learning. It is used to solve classification and regression problems.. Nonetheless, its main use is in Machine Learning Classification issues. The goal of the support vector machine method is to locate the ideal decision boundary or line in n-dimensional space, so that it may categories following data points in the most effective way possible. This ideal boundary for deciding between two options is defined by a hyperplane.To construct the hyperplane, SVM selects the most extreme points and vectors. The support vectors used in the Support Vector Machine method are the inspiration for the method's name.

C) LOGISTIC REGRESSION :

LR is a statistical technique that might be used to a challenge of classifying data. the likelihood of something happening by multiplying its log odds by some number of independent factors and then doing a linear mix. In statistical terms, binary logistic regression relies on a single dependent variable that may take on just two possible values, "0" or "1," as indicated by using either a binary or a continuous variable. Linear regression may be used to analyse a sigmoid function and identify its elements. The weight and bias of a linear function are its two adjusting variables.

D) NEURAL NETWORK :

A neural network is a collection of algorithms that attempts to replicate the manner in which the human brain processes data in order to find hidden connections within a given set of information. In this context, neural networks refer to systems that are composed of neurons and may originate either organically or synthetically. Because neural networks are able to adjust their behaviour in response to shifting inputs, the network may be able to provide the best possible output even if the criteria for that output have not been altered. When it comes to the design of trading systems, the idea of neural networks, which is based on artificial intelligence, is quickly becoming more and more relevant.

PERFORMANCE MATRICS

A) **ACCURACY:** How well our model predicts the appropriate class or labels is referred to as accuracy. This should be our benchmark measure for assessing the performance of our version if our dataset is reasonably balanced and all classes are of similar relevance.

$$\text{ACCURACY} = \frac{\text{TRUE POSITIVES} + \text{TRUE NEGATIVES}}{\text{ALL SAMPLES}}$$

B) **PRECISION :** We were able to determine how often we were right overall when predicting a monster's weakness by measuring the model's accuracy. However, we must employ the accuracy measure if we want to determine how often our predictions that a monster was weak against fire were accurate.

$$\text{PRECISION} = \frac{\text{TOTAL POSITIVES}}{\text{TOTAL PREDICTED POSITIVES}}$$

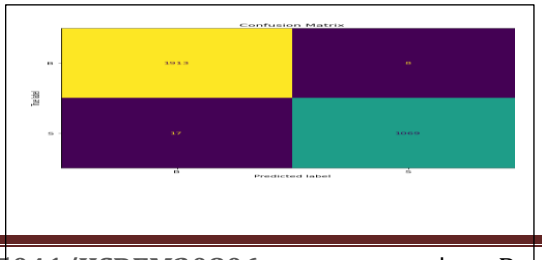
C) **RECALL :** The number of Positive samples that were properly identified is compared to the total number of Positive samples to determine recall. Recall measures the model's accuracy in properly identifying positive samples. The greater the recall, the better the accuracy, in other words.

$$\text{Recall} = \frac{\text{TRUE POSITIVE}}{\text{TRUE POSITIVE + FALSE NEGATIVE}}$$

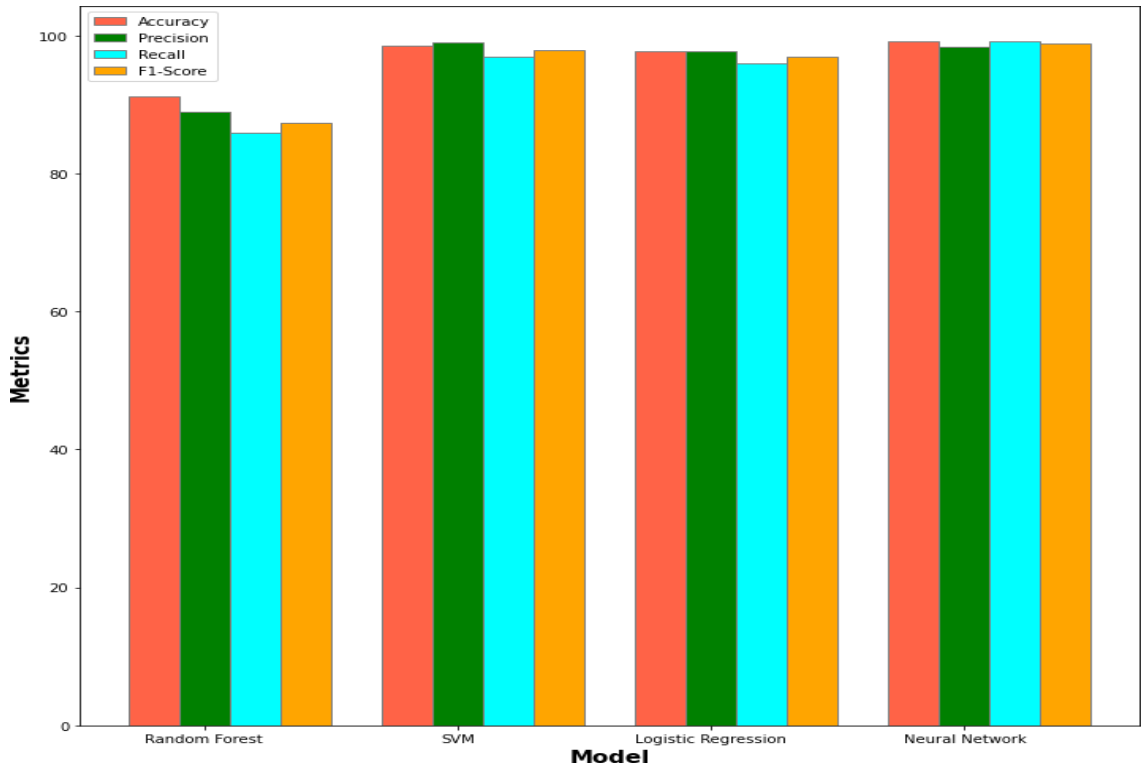
D) **F1 SCORE** : The F1 Score measures how well a binary classification model predicts the correct answer. Precision and Recall are used to arrive at this conclusion. That's a special sort of score that takes into account both accuracy and recall. That's why the F1 Score may be calculated by weighting accuracy and recall equally and averaging the two.

$$\text{F1-SCORE} = 2 * \frac{\text{PRECISION * RECALL}}{\text{PRECISION + RECALL}}$$

RESULTS :



MODEL	ACCURACY	PRECISION	RECALL	F1-SCORE
RANDOM FOREST	91.15397406052544	89.0279114533205	85.88672237697307	87.42911153119093
SUPPORT VECTOR MACHINE	98.57000332557367	99.05123339658444	96.93593314763231	97.98216799624589
LOGISTIC REGRESSION	97.80512138343865	97.82403027436139	96.00742804085422	96.90721649484536
NEURAL NETWORKS	99.16860658463585	98.43462246777163	99.25719591457754	98.84419787332408



CONCLUSION:

In this research, we offer a unique Android malware detection method that makes use of a large number of characteristics that are reflective of the qualities of apps in many ways. We conducted several tests as part of the examination. Different detection models were examined for their sensitivity. We also conducted an experiment to show how easily our detection model can be improved. To identify Android malware, we have utilized a variety of techniques, including Neural Networks, Logical Regression, Random Forest, and Support Vector Machines. Our team has presented four methods to enhance detection precision and facilitate malware identification. All four methods have been proved to have excellent detection accuracy in experiments. We also conducted tests to test the robustness against obfuscation and the viability of unsupervised learning-based categorization. Since then, our methodology has been adopted for use in Android malware detection, proving its efficacy in that setting.

IV. REFERENCES

[1] Kim, Tae Guen; Kang, Boo Joong; Rho, Mina; Sezer, Sakir; Im, Eul Gyu (2018). A Multimodal Deep Learning Method for Android Malware Detection using Various Features. IEEE Transactions on Information

Forensics and Security, (), 1–1. doi:10.1109/TIFS.2018.2866319

[2] Eom, Taehoon; Kim, Heesu; An, SeongMo; Park, Jong Sou; Kim, Dong Seong (2018). [IEEE 2018 International Conference on Software Security and Assurance (ICSSA) - Seoul, Korea (South) (2018.7.26- 2018.7.27)] 2018 International Conference on Software Security and Assurance (ICSSA) - Android Malware Detection Using Feature Selections and Random Forest. , (), 55–61. doi:10.1109/ICSSA45270.2018.00023

[3] Pang, J., & Bian, J. (2019). Android Malware Detection Based on Naive Bayes. 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS). doi:10.1109/icsess47205.2019.9040

[4] Rohit Srivastava;R.P. Mishra;Vivek Kumar;Himanshu Kumar Shukla;Neha Goyal;Chandrabhan Singh; (2020). Android Malware Detection Amid COVID-19 . 2020 9th International Conference System Modeling and Advancement in Research Trends (SMART), (), –. doi:10.1109/smart50582.2020.9337105

[5] K, Santosh Jhansi; Chakravarty, Sujata; Varma P, Ravi Kiran (2020). [IEEE 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI) - Tirunelveli, India (2020.6.15-2020.6.17)] 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184) - Feature Selection and Evaluation of Permission-based Android Malware Detection. , (), 795–799. doi:10.1109/ICOEI48184.2020.9142929

[6] Yuan, Wei; Jiang, Yuan; Li, Heng; Cai, Minghui

(2020). A Lightweight On-Device Detection Method for Android Malware. IEEE Transactions on Systems, Man, and Cybernetics: Systems, (), 1–12. doi:10.1109/TSMC.2019.2958382

[7] Ali Al Zaabi;Djedjiga Mouheb; (2020). Android Malware Detection Using Static Features and Machine Learning . 2020 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI), (), –. doi:10.1109/CCCI49893.2020.9256450

[8] Prerna Agrawal;Bhushan Trivedi; (2020). Evaluating Machine Learning Classifiers to detect Android Malware . 2020 IEEE International Conference for Innovation in Technology (INOCON), (), –. doi:10.1109/inocon50539.2020.9298290

[9] Ahmed Hashem El Fiky;Ayman Elshenawy;Mohamed Ashraf Madkour; (2021). Detection of Android Malware using Machine Learning . 2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), (), –. doi:10.1109/miucc52538.2021.9447661

[10] Iman Almomani, Aala Alkhayer and Walid El - Shafai "An Automated Vision-Based Deep Learning Model for Efficient Detection of Android Malware Attacks" IEEE Access Volume: 10 Page(s): 2700 - 2720 DOI: 10.1109/ACCESS.2022.3140341

[11] Ikram UL Haq, Tamim Ahmed Khan, and Adan Akhunzada "ADynamicRobustDL-BasedModelfor AndroidMalwareDetection" IEEEAccessVolume:9 Page(s): 74510 - 74521 DOI: 10.1109/ACCESS.2021.3079370

[12] Zhu, H., Li, Y., Li, R., Li, J., You, Z., & Song, H. (2021).SEDMDroid: An Enhanced Stacking Ensemble Framework for Android Malware Detection. IEEE Transactions on Network Science and Engineering, 8(2), 984–994. doi:10.1109/tnse.2020.2996379

[13] Hyun-Il Kim, Moonyoung Kang, Seong-Je Choand Sang-Il Choi "Efficient Deep Learning Network With Multi- Streams for Android Malware Family Classification" IEEE Access Volume: 10 Page(s): 5518 - 5532 DOI: 10.1109/ACCESS.2021.3139334