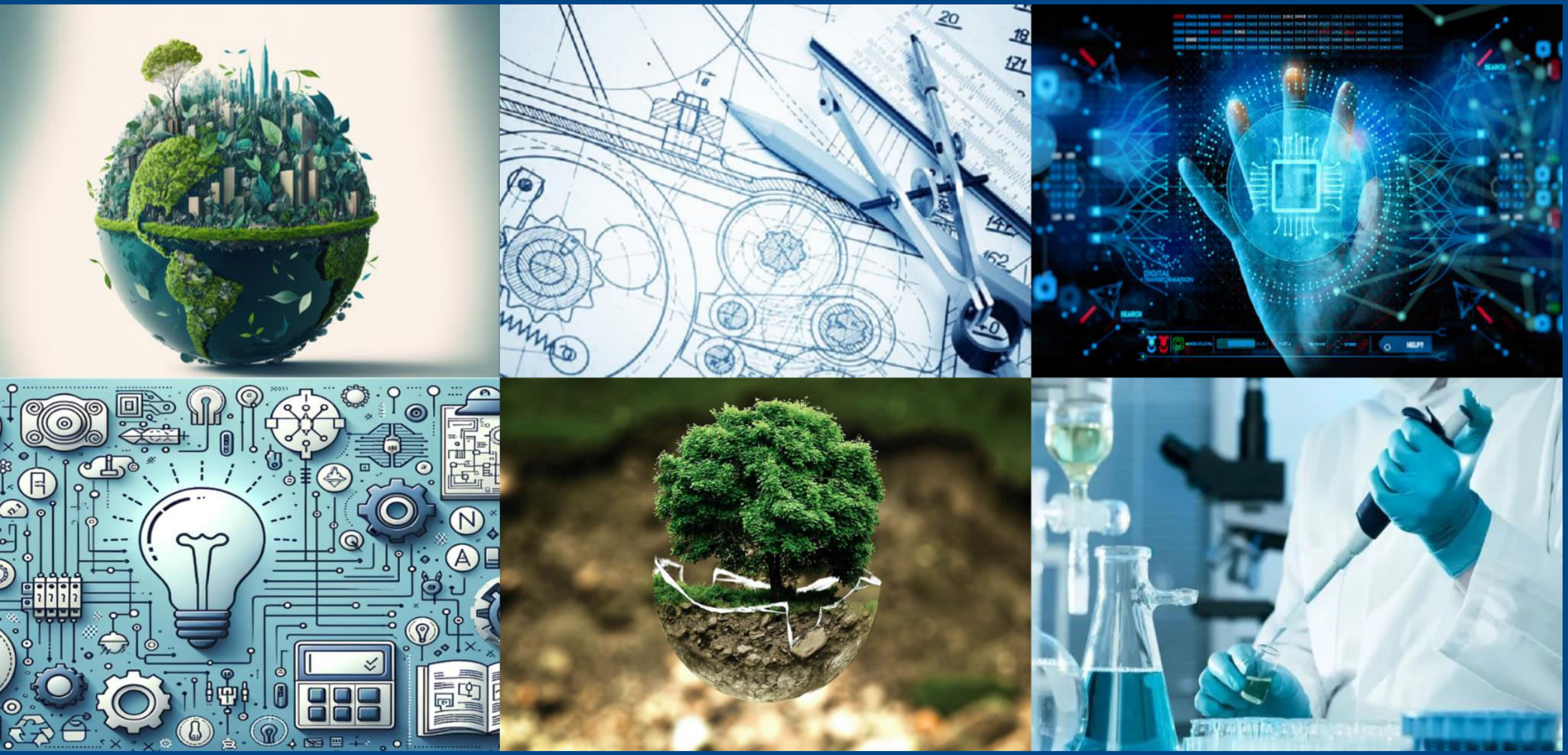
ISSN: 2582-7219

International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.206

Volume 8, Issue 3, March 2025

www.ijmrset.com ✉ ijmrset@gmail.com ☎ +91 6381 907 438 ☎ +91 63819 07438

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Malware Detection in Android using Machine Learning

Mr. Mande. Srinivasa Rao[1], K. Naga Venkata Durga[2], G. Chandini[3], B. Jyothi Vardhini[4], A. Ratnavalli[5]

Assistant Professor, Department of ECE, Vasireddy Venkatadri Institute of Technology, Nambur, Guntur, A.P., India[1]

Undergraduate Students, Department of ECE, Vasireddy Venkatadri Institute of Technology, Nambur, Guntur, A.P., India[2-5]

ABSTRACT: The Android is the world's most popular and widely used operating system for mobile smartphones today. One of the reasons for this popularity is the free third-party applications that are downloaded and installed. Unfortunately, this flexibility of installing any application created by third parties has also led to an endless stream of constantly evolving malware applications that are intended to cause harm to the user in many ways. In this project, different approaches for tackling the problem of Android malware detection are presented and demonstrated. The data analytics of a real-time detection system is developed. The detection system can be used to scan through installed applications to identify potentially harmful ones so that they can be uninstalled. This is achieved through machine learning models. Most Important Machine Learning algorithms are applied on the android application datasets and then we get to predict as detect android application contains malwares.

KEYWORDS: Malware detection, machine learning, malicious, benign.

I. INTRODUCTION

The main purpose behind our project "MALWARE DETECTION IN ANDROID USING MACHINE LEARNING" is despite the growing threat of malware, there is still no reliable and robust method for detecting malicious applications. However, with the increasing use of machine learning in various fields, we believe that this issue can be addressed through the application of machine learning techniques. Our project aims to conduct a thorough and systematic investigation into the use of machine learning for malware detection, with the ultimate goal of developing an efficient ML model capable of accurately classifying apps as either **benign (0)** or **malware (1)** based on their requested permissions.

II. SYSTEM ARCHITECTURE

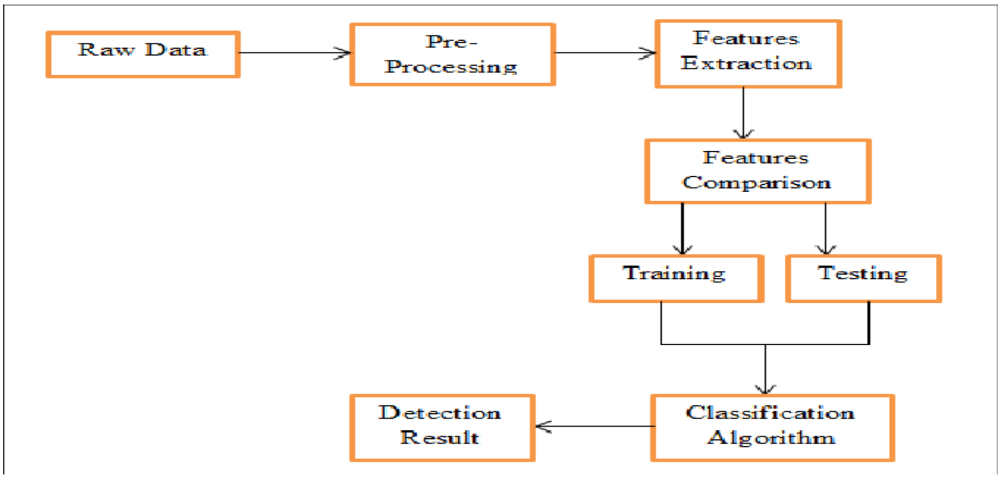


Fig 1.1 System architecture

Data Collection: The first step involves gathering Android app data that is both benign and malicious.
Feature Extraction: Extract relevant features from the collected app data. This could include static features (like app permissions, manifest file analysis) and dynamic features (like system calls, memory behavior).
Data Preprocessing: Preprocess the data by cleaning it (removing irrelevant features or duplicate data), normalizing the data (scaling features to a common range), and transforming the data into a format suitable for machine learning algorithms.
Train/Test Data Split: Divide the data into two subsets: a training set and a testing set. Typically, the data is split into 80% for training and 20% for testing, although this ratio can vary.
Model Selection: Choose an appropriate machine learning model based on the type of data and the problem.
Common models used in malware detection include:

- **Support Vector Machines (SVM)**: Great for classification tasks.
- **Decision Trees**: Useful for identifying patterns based on features.
- **Neural Networks**: Especially deep learning models are effective when handling large datasets with complex patterns.

Malware Detection: The deployed model analyzes each app for potential threats based on the features learned during training.
Alert User: If malware is detected, the system sends an alert to the user, informing them of the detected threat, and possibly suggests actions like quarantine or removal of the app.

III. ALGORITHMS USED

Machine learning algorithms: Such as K-Nearest Neighbor (KNN), Decision Tree, Random Forest, SVM, Logistic, and Regression. Machine learning algorithms analyze strings from files to learn patterns that distinguish between malicious and benign strings.

Dynamic analysis techniques: These techniques are used to increase code coverage and improve malware detection performance.

Genetic algorithm module: This module optimizes feature selection, providing a subset of relevant features to the machine learning module.

Network monitoring: Monitoring the network on which the Android device is connected can help identify suspicious applications.

MODULES USED :-

TENSORFLOW:

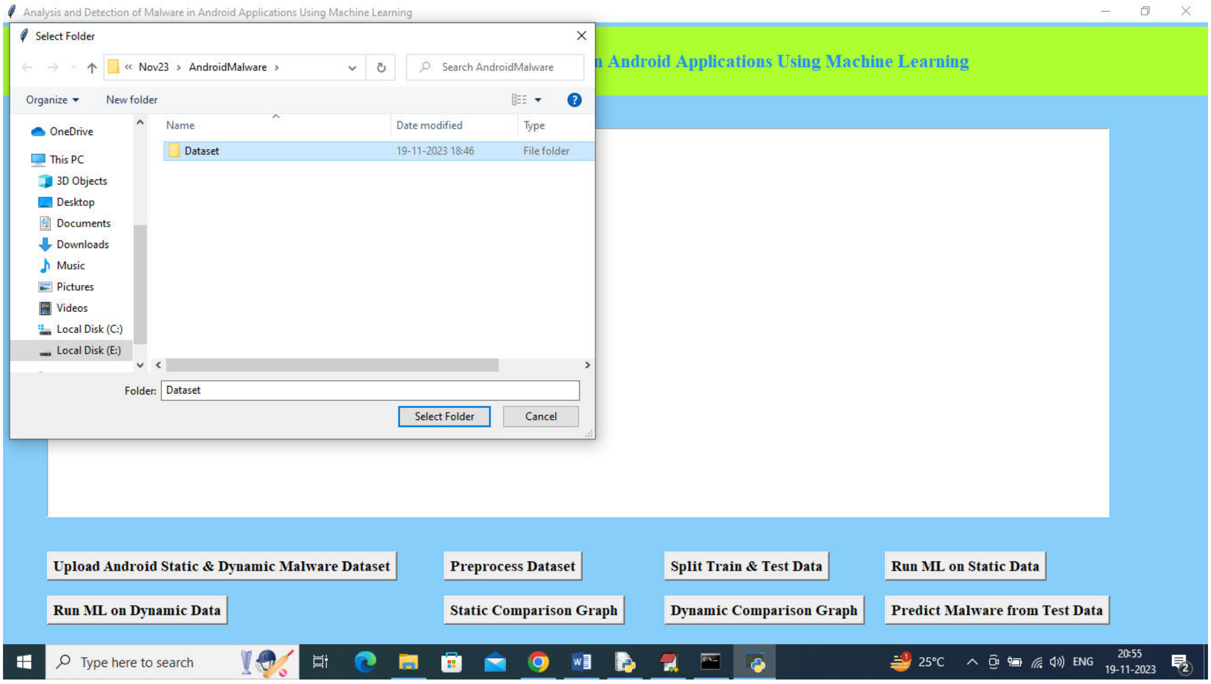
TensorFlow is an open-source machine learning library developed by Google. It provides a comprehensive set of tools and functionalities for building and training machine learning models, particularly deep learning models. TensorFlow allows users to define computational graphs and execute them efficiently across multiple devices, including CPUs, GPUs, and TPUs (Tensor Processing Units). It offers a wide range of pre-built neural network layers and modules, making it easier to design complex architectures.

Scikit – learn:

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

IV. RESULTS AND DISCUSSION

TESTING THE OUTPUT:-



Dataset 19-11-2023 18:46 File folder

Folder: Dataset

Fig 2.1 Model interface

In above screen click on “Upload Android Static & Dynamic Malware Dataset" button to upload dataset and then read and display both dataset values. In above screen selecting and displaying entire ‘Dataset’ folder with static and dynamic and then click on ‘Select Folder” button to load datasets

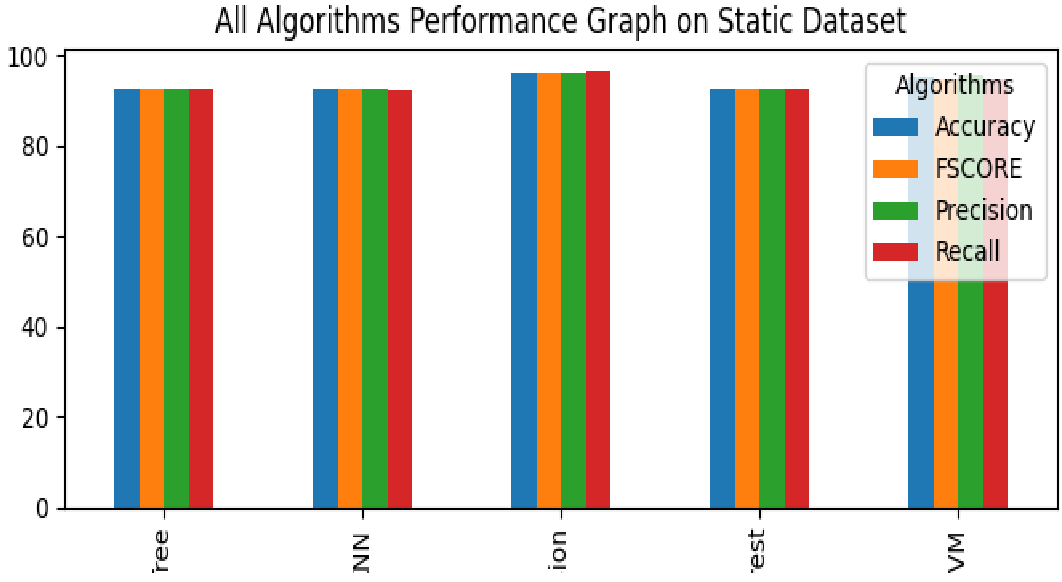


Fig 2.2 Algorithms graphs on static dataset

© 2025 IJMRSET | Volume 8, Issue 3, March 2025| DOI: 10.15680/IJMRSET.2025.0803325

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

In above graph can see performance of all algorithms on static dataset where x-axis represents algorithm names and y-axis represents accuracy and other metrics in different colour bars and in all algorithms we can see Random Forest and Decision Tree got high performance. Now click on 'Dynamic Comparison Graph' button to get below Dynamic graph.

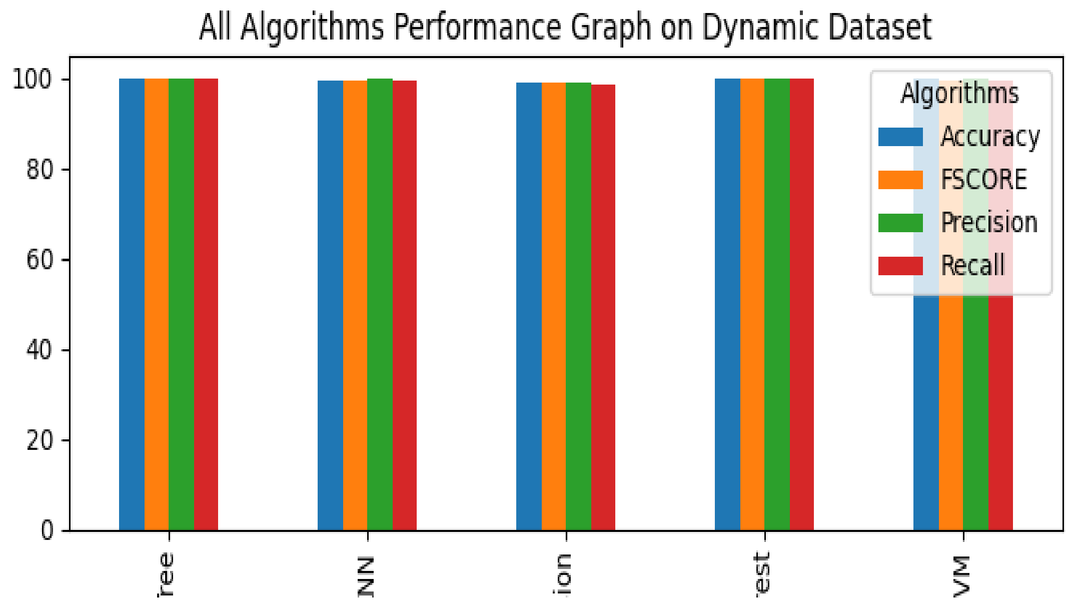
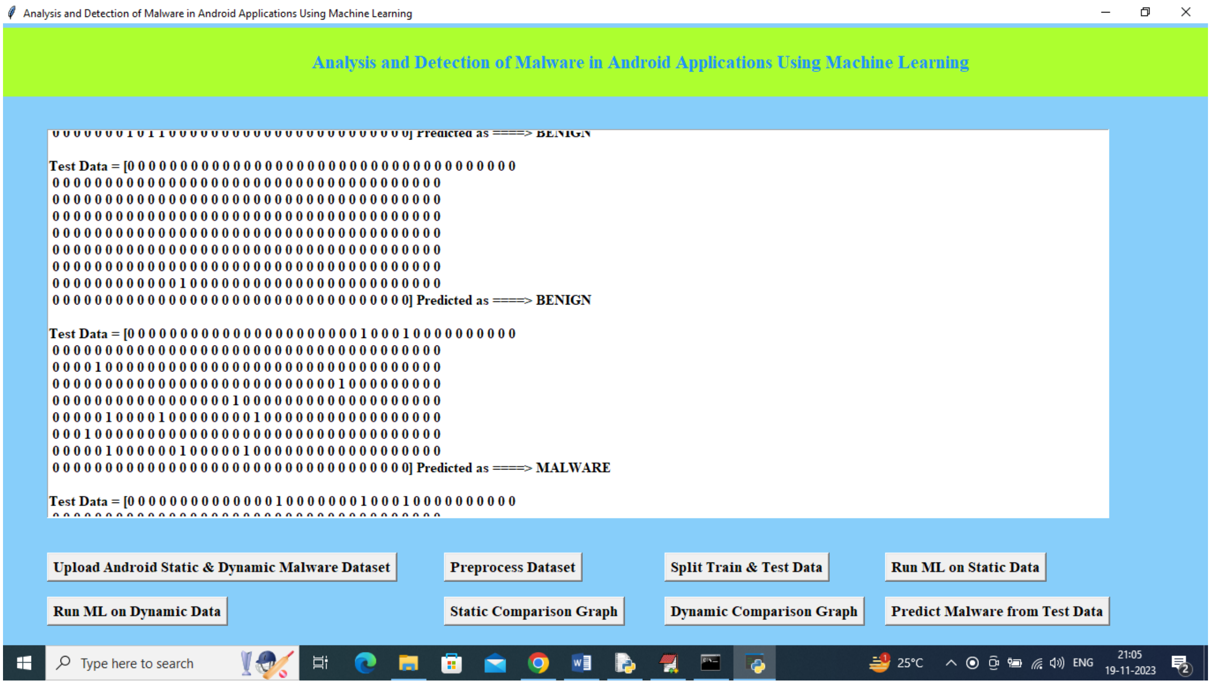


Fig 2.3 Algorithms performance Graph on Dynamic dataset

FINAL RESULTS:-


Analysis and Detection of Malware in Android Applications Using Machine Learning
Analysis and Detection of Malware in Android Applications Using Machine Learning

```
0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] Predicted as ====> BENIGN

Test Data = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] Predicted as ====> BENIGN

Test Data = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] Predicted as ====> MALWARE

Test Data = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Fig 2.4 Final output

In above screen selecting and uploading 'testData.csv' file and then click on 'Open' button to get below prediction output, in square bracket we can see Test Data values and after arrow symbol we can see predicted output as Benign or Malware.

V. CONCLUSION

Android malware detection using machine learning has emerged as an effective approach to counter evolving cybersecurity threats. Traditional signature-based detection methods struggle against polymorphic and zero-day malware, whereas machine learning models leverage static and dynamic analysis techniques to enhance detection accuracy. By analyzing permissions, API calls, system calls, and network behaviours, classifiers such as Support Vector Machines (SVM), Random Forest, and deep learning models provide robust malware identification. However, challenges such as adversarial attacks, feature selection, and scalability remain. The integration of hybrid models combining static and dynamic analysis can improve accuracy while reducing false positives. Future research should focus on explainability, real-time detection, and optimization for resource-constrained mobile devices. In conclusion, machine learning-driven Android malware detection represents a promising direction for strengthening mobile security, requiring continuous advancements to combat sophisticated cyber threats effectively.

REFERENCES

1. **Arp, Daniel, et al.**
   "DREBIN: Effective and explainable detection of Android malware in your pocket." https://www.ndss-symposium.org/ndss2014/drebin-effective-and-explainable-detection-android-malware-your-pocket
2. **Sahs, Justin, and Latifur Khan.**
   "A machine learning approach to Android malware detection using application permissions." https://ieeexplore.ieee.org/document/6413730
3. **Aafer, Yousra, Wenliang Du, and Heng Yin.**
   "DroidAPIMiner: Mining API-level features for robust malware detection in Android." https://dl.acm.org/doi/10.1007/978-3-319-04283-1_2
4. **Zhang, Yuchen, et al.**
   "Enhancing Android malware detection with deep learning." https://ieeexplore.ieee.org/document/8717303
5. **Wu, Yiming, et al.**
   "Android malware detection based on system call sequences and machine learning." https://www.sciencedirect.com/science/article/pii/S0167404820301444
6. **Li, Ziming, et al.**
   "Android malware detection based on a hybrid deep learning model." https://www.sciencedirect.com/science/article/pii/S0167404819302252
7. **Yerima, Suleiman Yahaya, and Sakir Sezer.**
   "Machine learning for Android malware detection using permission and API call features." https://www.tandfonline.com/doi/abs/10.1080/19393555.2015.1124340

INTERNATIONAL JOURNAL OF

MULTIDISCIPLINARY RESEARCH

IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com