***ISBN:978-93-48505-71-2***

# AN EFFECTIVE MACHINE LEARNING AND DEEP LEARNING ALGORITHMS FOR ANDROID MALWARE CLASSIFICATION AND DETECTION

**Ms.Sruthi K M, Assistant Professor, Department of Computer Science, Sree Narayana Guru College, Coimbatore, Tamilnadu, India.*
***Ms.Saranya S, Assistant Professor, Department of Computer Science, Sree Narayana Guru College, Coimbatore, Tamilnadu, India.*

## Abstract

With the growing reliance on mobile devices, malware attacks are becoming more frequent, particularly on Android phones, which hold 72.2% of the market share. Cybercriminals target smart phones using various tactics, including credential theft, surveillance, and malicious advertising. This paper presents a systematic review of machine learning-based techniques for detecting Android malware. Malware classification involves grouping malware into families based on their unique signature. This dataset is collected from kaggle repository. One of the most significant challenges facing mobile users today is malware. This study focuses on classifying emerging malware based on shared features of similar malware. It introduces a novel framework that categorizes malware samples into families and detects new malware for further analysis. To achieve this, Artificial Neural Network (ANN) technique is utilized. The proposed approach aids in identifying and eliminating new malware while effectively classifying them into their respective families. The Proposed ANN techniques are obtained the accuracy of 98%, which is 4% higher than the RF, 2% higher than the J48, and 3%, 1% higher than the SVM, and Navie Bayes.

**Keywords:** Machine Learning, Deep Learning, Artificial Neural Network (ANN), Malware, Classification, Detection, Android.

## Introduction

Google Android holds nearly 75% of the Smartphone market share, with 2.8 billion active users. It is open-source nature and ease of customization at various attracted users and encouraged manufacturers to produce low-cost smart devices. Additionally, Android is highly popular among developers due to its Software Development Kit (SDK), which enables them to create applications with minimal effort to reach a vast audience.

However, due to its large user base and widespread popularity, the activity of attackers targeting Android devices through malware apps has significantly increased in recent years. According to AV-Test data, the total number of malware instances has surged from 182 million to 1,342 million over the past decade. Malware apps are designed to perform unwanted actions, compromising users' privacy and sensitive data. Therefore, it is crucial to develop an efficient method to distinguish malware apps from legitimate ones. This technology can be leveraged to develop models that assess incoming data, identify anomalies, and detect malware. In Android malware detection, machine learning models analyze various features, such as requested permissions, API calls, and network activity, to identify patterns associated with malicious behaviour. Once trained, the model can classify new applications as either malicious or non-malicious. The results indicate that Random Forest and Ensemble techniques achieved the highest accuracy in malware detection. Performance evaluation was conducted using 2-fold cross-validation, F1-score, Precision, and Recall metrics.

*ISBN:978-93-48505-71-2*


**Figure: Types of cyber attacks**

Cyber-attacks have become one of the most critical concerns in modern technology. The term refers to the exploitation of system vulnerabilities for malicious purposes, such as data theft, system manipulation, or destruction. The growing threat of Android malware has significantly increased the need for analyzing prominent malware samples. Extensive research has been conducted on both static and dynamic malware analysis, utilizing static features and API calls, respectively. Malware detection modules analyze collected data, leveraging trained models to determine whether a specific application or network connection poses a security risk. For instance, a machine learning system can explicitly identify underlying patterns based on observed data.

## Proposed Methodology

The goal is to classify malware into families and detect new malware samples effectively. The methodology is divided into four stages. In first, malware dataset is taken from kaggle repository and the dataset is preprocessed in second stage. The preprocessed data is used for model construction at third stage and finally the model evaluated by the key metrics such as accuracy, precision, recall, f1-score, and confusion matrix, as illustrated in (Figure 1) flow diagram below.

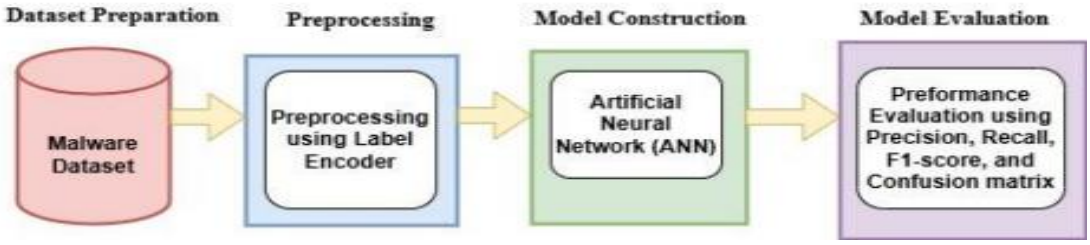


**Figure 1: Proposed Methodology flow**

### Dataset Preparation

1. Load benign Android application samples.
2. Load malicious Android application samples.
3. Return the collected benign and malicious samples.
4. Clean the benign samples to remove noise and irrelevant data.
5. Clean the malicious samples to remove noise and irrelevant data. 6. Return the cleaned benign and malicious samples.

### Data Preprocessing

1. Extract relevant features (e.g., transact, on Service Connected, android.os.Binder) from the cleaned benign samples.
2. Extract relevant features from the cleaned malicious samples.

*ISBN:978-93-48505-71-2*

3. Return the extracted features for both benign and malicious samples. 4. Select the most relevant features from the extracted benign and malicious features. 5. Return the selected features for further processing.

**Model Construction**

1. Split the selected features into training and testing datasets.
2. Initialize machine learning models (Random Forest, SVM, ANN, Naive Bayes, J48, and Ensemble).
3. Train each model using the training dataset.
4. Return the trained models and the testing dataset for evaluation.

**Model Evaluation**

1. For each trained model, predict the labels for the testing dataset.
2. Calculate and store the evaluation results for each model.
3. Return the evaluation results.

**Malware Prediction**

1. Load new, unseen Android application samples.
2. Extract features from the new samples.
3. Use the trained models to predict whether the new samples are malicious or benign. 4. Return the predictions for the new samples.

**Results and Discussion**

The results of the experiments are summarized in the table below. The proposed algorithm (Artificial Neural Network - ANN) is compared with Navie Byes, J48, Ensemble, and SVM. Figure 2 depicts the two targeted class labels such as Benign (0) and S (1- malicious/ suspicious). Malicious samples are limited than the benign samples. The bar chart represents the class balance of a malware dataset, showing the distribution of two classes: 'B' (Benign) and 'S' (Suspicious/Malware)**.** The y-axis (Count) indicates the number of samples in each class, while the x-axis (Classes) represents the two categories. From the visualization, it is evident that the dataset is imbalanced, with the 'B' class having a significantly higher number of samples than the 'S' class. Such imbalance can negatively impact machine learning models, leading to biased predictions where the model favors the majority class. To address this, techniques such as oversampling (e.g., SMOTE), under sampling, or class-weight adjustment can be applied to improve classification performance and ensure better malware detection.

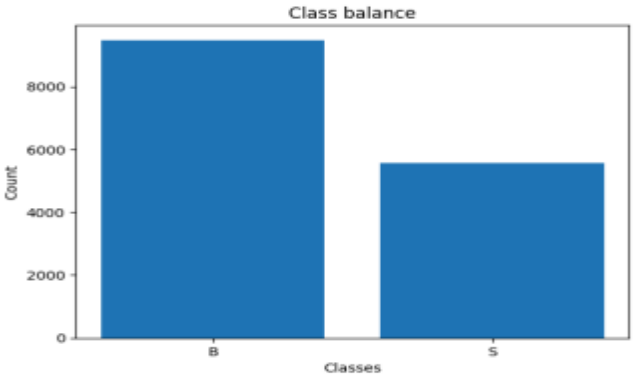


**Figure 2: Imbalance dataset**

*ISBN:978-93-48505-71-2*

**Figure 3 Accuracy and Loss of Train and Test set**

Figure 3 plots the training progress of a malware detection model over five epochs. It shows the increase in training accuracy and validation accuracy over time. Both curves indicate a steady improvement, suggesting that the model is learning effectively. The Loss plot depicts the training loss and validation loss, both of which decrease over epochs, indicating proper convergence of the model. However, a slight fluctuation in validation loss around epoch 3 suggests minor overfitting or noise in the data.

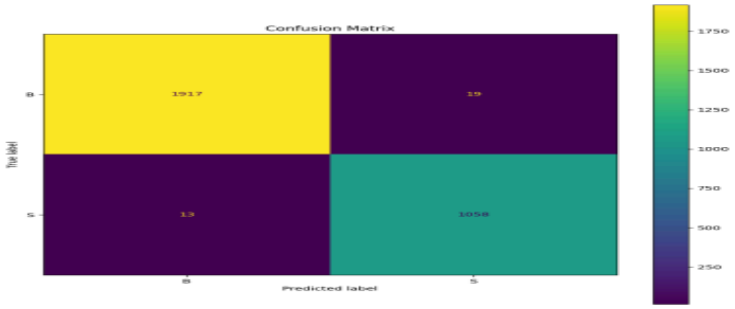


**Figure 3: Confusion matrix of proposed ANN**

The (Figure 3) confusion matrix represents the performance of a malware detection model by comparing actual labels with predicted labels. The two classes are 'B' (Benign) and 'S' (Suspicious/Malware). The matrix shows that 1,917 benign samples were correctly classified as benign **(**True Positives**)** and 1,058 malware samples were correctly classified as malware **(**True Negatives**)**.

**References**


- Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2017, September). Deep android malware detection and classification. In *2017 International conference on advances in computing, communications and informatics (ICACCI)* (pp. 1677-1683). IEEE.
- Elayan, O. N., & Mustafa, A. M. (2021). Android malware detection using deep learning. *Procedia Computer Science*, *184*, 847-852.
- Su, X., Zhang, D., Li, W., & Zhao, K. (2016, August). A deep learning approach to android malware feature learning and detection. In *2016 IEEE Trustcom/BigDataSE/ISPA* (pp. 244-251). IEEE.
- Imtiaz, S. I., ur Rehman, S., Javed, A. R., Jalil, Z., Liu, X., & Alnumay, W. S. (2021). DeepAMD: Detection and identification of Android malware using high-efficient Deep Artificial Neural Network. *Future Generation computer systems*, *115*, 844-856.
- Yadav, P., Menon, N., Ravi, V., Vishvanathan, S., & Pham, T. D. (2022). A two-stage deep learning framework for image-based android malware detection and variant classification. *Computational Intelligence*, *38*(5), 1748-1771.