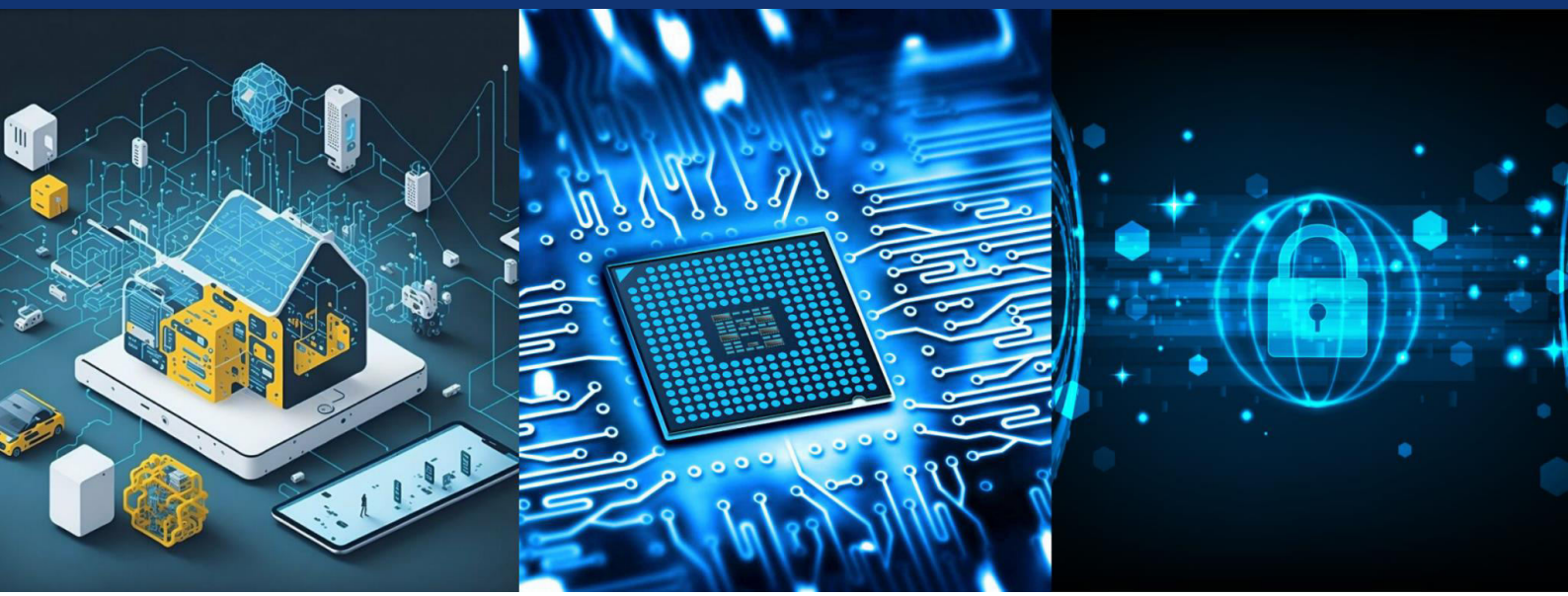
IJIRCCE

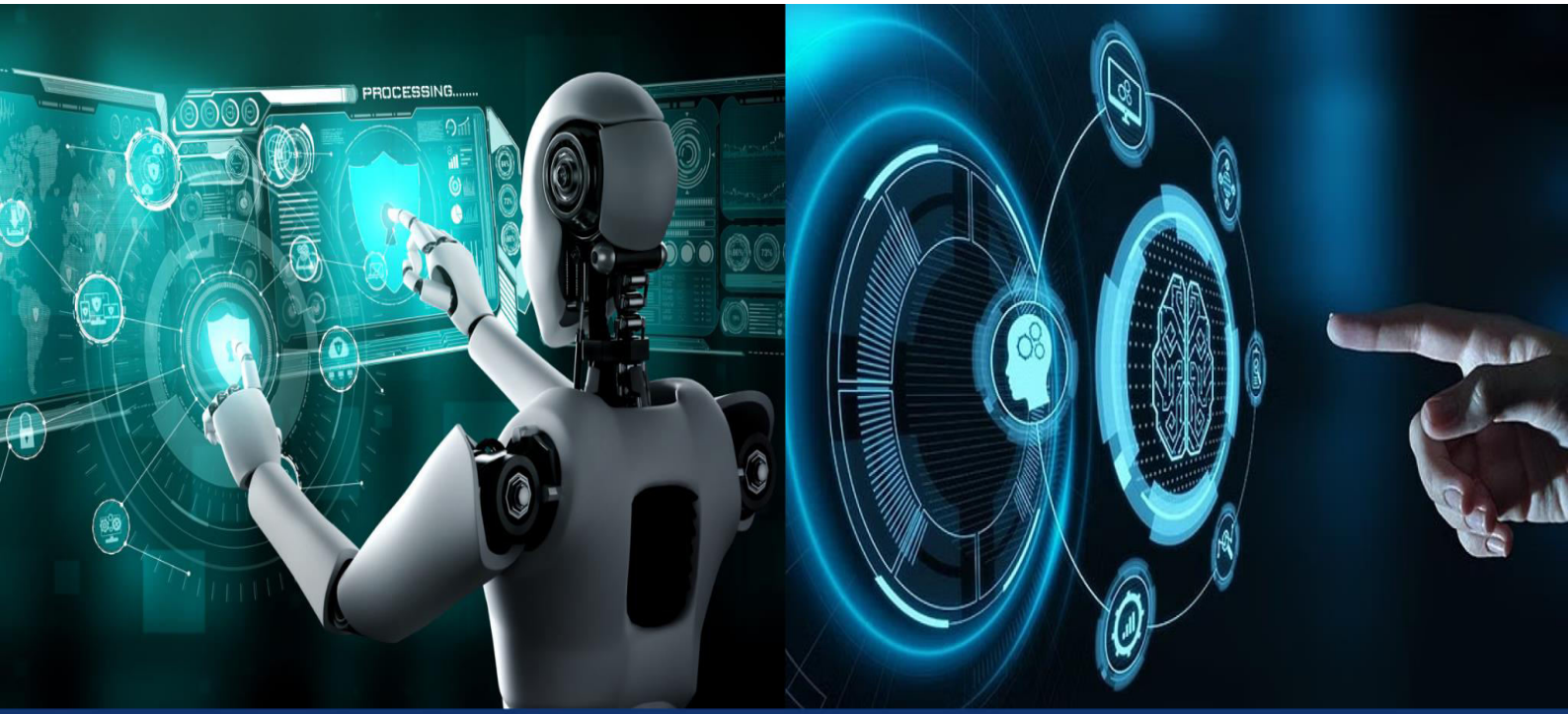
ISSN(O): 2320-9801

ISSN(P): 2320-9798



# International Journal of Innovative Research in Computer and Communication Engineering

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




**Impact Factor: 9.927**

**Volume 14, Issue 8, August 2026**

© 2026 IJIRCCE | Volume 14, Issue 8, August 2026 DOI: 10.15680/IJIRCCE.2026.1408015

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

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

# Machine Learning based Hybrid Approach for Prediction Model of Android Malware

**Raj Sagar Kumar, Dr. Ritesh Kumar Yadav**

M. Tech Scholar, Department of CSE, Sarvepalli Radhakrishnan University, Bhopal, India

Professor, Department of CSE, Sarvepalli Radhakrishnan University, Bhopal, India

**ABSTRACT:** Android malware continues to evolve in sophistication, rendering traditional signature-based detection methods increasingly inadequate for identifying novel and obfuscated threats. This study proposes a hybrid machine learning approach that combines Support Vector Machines (SVM) and Multi-Layer Perceptrons (MLP) to construct a robust prediction model for Android malware detection. The outputs of both classifiers are integrated using an ensemble or stacking strategy, where predictions are fused to improve overall classification accuracy, reduce false positive rates, and enhance detection of previously unseen malware variants. Experimental evaluation on benchmark Android malware datasets demonstrates that the proposed SVM-MLP hybrid model achieves superior performance in terms of accuracy, precision, recall, and F1-score compared to standalone SVM or MLP classifiers, underscoring the potential of hybrid machine learning architectures for building more resilient and adaptive Android malware prediction systems.

**KEYWORDS:** Android, Hybrid classfier, Malware, Artificial Intelligence, Security, Attack, Cyber.

## I. INTRODUCTION

The rapid proliferation of smartphones has fundamentally transformed the way individuals communicate, work, and access information, with Android emerging as the dominant mobile operating system worldwide [1]. This widespread adoption, however, has made Android devices an attractive target for cybercriminals, resulting in an alarming surge in malicious applications designed to compromise user privacy, steal sensitive data, and disrupt system functionality [2]. As the Android ecosystem continues to expand through its open-source architecture and flexible application distribution channels, the volume and complexity of malware variants have grown correspondingly, posing significant challenges to conventional security mechanisms [3].

Traditional malware detection techniques have primarily relied on signature-based methods, which compare application code against a database of known malware signatures [4]. While effective against previously identified threats, these approaches are inherently limited in their ability to detect zero-day attacks and polymorphic malware that continuously alter their code structure to evade detection [5]. Furthermore, the increasing use of obfuscation techniques, such as code encryption, dynamic loading, and reflection, has rendered static signature matching increasingly insufficient for identifying sophisticated malicious applications [6]. These limitations have motivated researchers to explore more adaptive and intelligent detection paradigms capable of generalizing beyond known malware patterns.

Machine learning has emerged as a promising alternative for malware detection, offering the capability to learn discriminative patterns from large-scale datasets and generalize to previously unseen threats [7]. By analyzing static features such as permissions, API calls, and intent filters, or dynamic features such as system call sequences and network behavior, machine learning models can construct predictive frameworks that distinguish malicious applications from benign ones with considerably higher accuracy than rule-based systems [8]. Among the various algorithms explored in this domain, Support Vector Machines (SVM) have demonstrated strong performance in high-dimensional feature spaces, owing to their ability to identify optimal decision boundaries even with relatively limited training data [9]. However, SVM models often struggle to capture highly non-linear and complex relationships that are characteristic of modern malware behavior.

To address this limitation, Multi-Layer Perceptrons (MLP), a class of feedforward artificial neural networks, have been increasingly adopted for malware classification tasks due to their capacity to model intricate, non-linear decision

© 2026 IJIRCCE | Volume 14, Issue 8, August 2026 DOI: 10.15680/IJIRCCE.2026.1408015

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

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

boundaries through multiple layers of interconnected neurons [10]. Despite their strong representational power, MLPs are prone to overfitting when trained on limited or imbalanced datasets and often require extensive hyperparameter tuning to achieve optimal performance [11]. Given the complementary strengths and limitations of SVM and MLP, researchers have begun investigating hybrid architectures that combine multiple classifiers to leverage their respective advantages while mitigating individual weaknesses, thereby improving overall detection robustness and generalization [12].

Motivated by these developments, this study proposes a hybrid machine learning framework that integrates SVM and MLP for Android malware prediction. The rationale behind this hybridization lies in exploiting SVM's strong margin-based classification capability alongside MLP's proficiency in learning complex non-linear feature interactions, resulting in a more comprehensive and accurate detection model. By combining static feature extraction with an ensemble-based classification strategy, the proposed approach aims to enhance detection accuracy, reduce false positive rates, and improve resilience against evolving and obfuscated malware threats. The remainder of this paper is organized as follows: related work on Android malware detection techniques is reviewed, followed by a detailed description of the proposed hybrid SVM-MLP methodology, experimental setup and dataset description, results and performance evaluation, and finally, concluding remarks along with directions for future research.

## II. PROPOSED METHODOLOGY

The proposed hybrid Android malware prediction framework follows a systematic pipeline comprising dataset acquisition, preprocessing, feature reduction, classification, and result generation, as illustrated in the accompanying flowchart. Each stage is described in detail below, along with the mathematical formulations underlying the classification and hybridization processes.

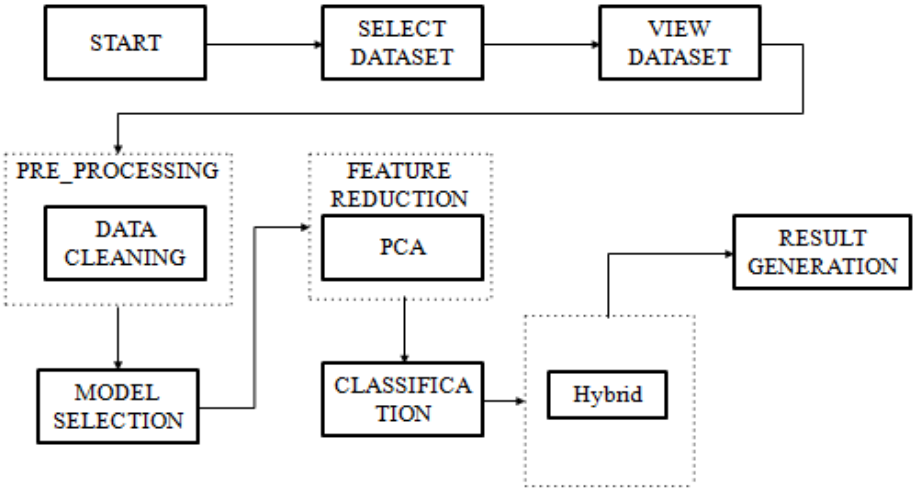



Figure 1: Flow Chart

The proposed hybrid Android malware prediction framework follows a systematic pipeline comprising dataset acquisition, preprocessing, feature reduction, classification, and result generation, as illustrated in the accompanying flowchart. Each stage is described in detail below, along with the mathematical formulations underlying the classification and hybridization processes.

**Dataset Selection and Viewing**

The first stage involves selecting an appropriate Android malware dataset comprising labeled samples of benign and malicious applications. Each sample is represented as a feature vector $x_i \in \mathbb{R}^n$, extracted from static attributes such as permissions, API calls, intents, and opcode sequences, along with a corresponding class label $y_i \in \{0, 1\}$, where $0$ denotes a benign application and $1$ denotes malware. Once selected, the dataset is viewed and inspected to understand its structure, class distribution, and potential inconsistencies before further processing.

© 2026 IJIRCCE | Volume 14, Issue 8, August 2026 DOI: 10.15680/IJIRCCE.2026.1408015

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

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

**Preprocessing**

Raw datasets typically contain noise, missing values, redundant features, and inconsistent formatting, all of which can adversely affect model performance. The preprocessing stage therefore performs data cleaning, which includes handling missing values, removing duplicate entries, and normalizing feature values to a common scale. Normalization is performed using min-max scaling:

$$x_i' = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$

where $x_i$ is the original feature value, and $x_{min}$, $x_{max}$ are the minimum and maximum values of that feature across the dataset. This ensures that all features contribute proportionally during model training, preventing features with larger numeric ranges from dominating the learning process.

**Model Selection**

Following data cleaning, an appropriate base learning strategy is selected for subsequent classification. This stage involves identifying suitable candidate algorithms—in this case, Support Vector Machine (SVM) and Multi-Layer Perceptron (MLP)—based on their theoretical suitability for the malware detection problem, taking into account the dataset's dimensionality, size, and class balance. This selection informs how the classification stage will later be structured.

**Feature Reduction using PCA**

Given the high dimensionality of extracted Android application features, Principal Component Analysis (PCA) is applied to reduce the feature space while retaining maximum variance in the data. PCA transforms the original correlated feature set into a smaller set of uncorrelated principal components. Given a standardized data matrix $X \in \mathbb{R}^{m \times n}$, the covariance matrix is computed as:

$$C = \frac{1}{m-1} X^T X$$

Eigenvectors $v_k$ and eigenvalues $\lambda_k$ of $C$ are then computed, satisfying:

$$C v_k = \lambda_k v_k$$

The top $k$ eigenvectors corresponding to the largest eigenvalues are selected to form the projection matrix $W \in \mathbb{R}^{n \times k}$, and the reduced feature representation is obtained as:

$$X_{reduced} = XW$$

This dimensionality reduction step decreases computational complexity and mitigates the risk of overfitting during classification, while preserving the most informative variance in the dataset.

**Hybrid Classification (SVM + MLP)**

The reduced feature set is passed into the classification stage, where both SVM and MLP are trained independently and subsequently combined into a hybrid model.

**Support Vector Machine (SVM):** SVM constructs an optimal separating hyperplane that maximizes the margin between the two classes. Given training samples $(x_i, y_i)$, the decision function is defined as:

$$f_{SVM}(x) = \text{sign}\left(\sum_{i=1}^{m} \alpha_i y_i K(x_i, x) + b\right)$$

where $\alpha_i$ are Lagrange multipliers, $b$ is the bias term, and $K(x_i, x)$ is a kernel function (commonly the Radial Basis Function) defined as:

$$K(x_i, x) = \exp(-\gamma \parallel x_i - x \parallel^2)$$

© 2026 IJIRCCE | Volume 14, Issue 8, August 2026 DOI: 10.15680/IJIRCCE.2026.1408015

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

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

**Multi-Layer Perceptron (MLP):** MLP consists of an input layer, one or more hidden layers, and an output layer, with each neuron applying a weighted sum followed by a non-linear activation function. For a hidden layer, the output is computed as:

$$h_j = \sigma\left(\sum_{i=1}^{n} w_{ij}\, x_i + b_j\right)$$

where $w_{ij}$ are the connection weights, $b_j$ is the bias, and $\sigma(\cdot)$ is the sigmoid activation function, $\sigma(z) = \frac{1}{1+e^{-z}}$. The final output layer produces the predicted class probability:

$$\hat{y}_{MLP} = \sigma\left(\sum_{j=1}^{h} w_j\, h_j + b_o\right)$$

**Hybridization:** The outputs of both classifiers are combined using a weighted ensemble strategy to leverage the margin-based robustness of SVM and the non-linear learning capability of MLP. The hybrid decision score is computed as:

$$P_{hybrid}(x) = \beta \cdot f_{SVM}(x) + (1-\beta) \cdot \hat{y}_{MLP}(x)$$

where $\beta \in [0, 1]$ is a weighting coefficient that controls the relative contribution of each classifier, typically determined empirically through cross-validation. The final classification decision is obtained by thresholding the hybrid score:

$$\hat{y}_{final} = \begin{cases} 1 & \text{if } P_{hybrid}(x) \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

**Result Generation**

In the final stage, the predictions generated by the hybrid model are evaluated against ground-truth labels using standard performance metrics, including accuracy, precision, recall, and F1-score. The results are compiled and analyzed to assess the effectiveness of the hybrid SVM-MLP approach relative to standalone classifiers, forming the basis for the experimental evaluation presented in subsequent sections.

## III. SIMULATION RESULTS

The proposed work is simulated using the Spyder IDE within the Python 3.7 environment.

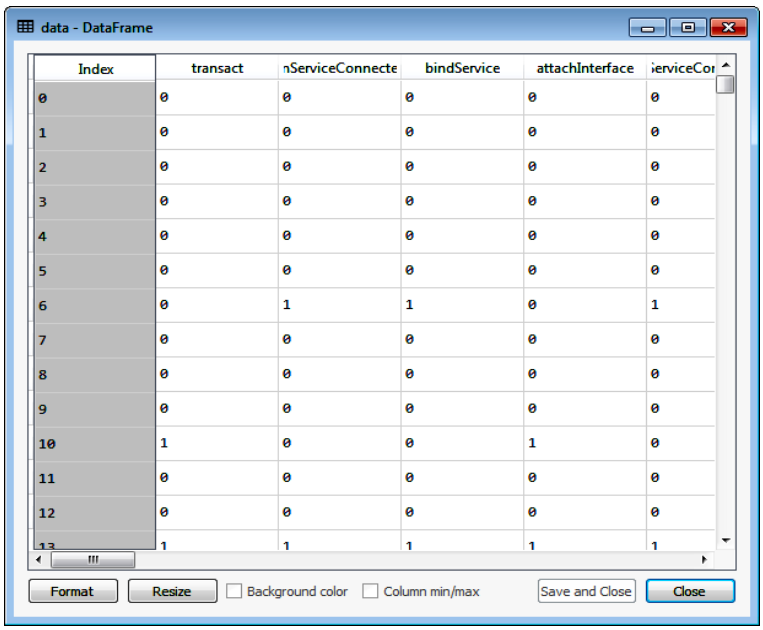



Figure 2: Dataset matrix

© 2026 IJIRCCE | Volume 14, Issue 8, August 2026 DOI: 10.15680/IJIRCCE.2026.1408015

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

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

Figure 2 is presenting the dataset matrix, in this dataset total 15036 number of rows and 215 number of column. In entire dataset, 5560 malware dataset and 9476 dataset benign or non malware.


Figure 3: Data train

Figure 3 is presenting the x train dataset. The information preparing system is the main part of any viable malware expectation model for Android.


Figure 4: Data Test

© 2026 IJIRCCE | Volume 14, Issue 8, August 2026 DOI: 10.15680/IJIRCCE.2026.1408015

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

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

This is the x test dataset, which is displayed in Figure 4. Following the finishing of the preparation interaction on a complete dataset, the Android malware expectation model is presently prepared to scrutinize its recently procured information.

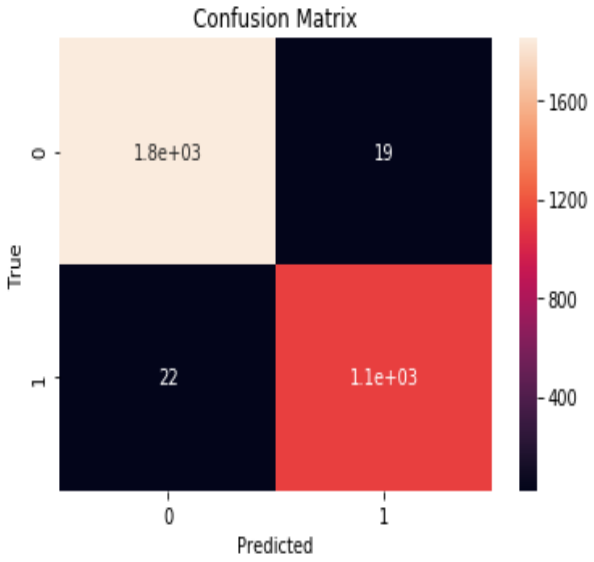



Figure 5: Confusion Matrix(CM)

Figure 5 is showing the Confusion Matrix(CM) of the proposed prediction model.

Table 1: Result Comparison

| Sr No | Method | Accuracy (%) | Precision (%) | Recall (%) | F_Measure (%) | Classification Error (%) |
|---|---|---|---|---|---|---|
| 1 | Proposed (Hybrid (SVM+MLP) | 99.60 | 99.45 | 98.87 | 99.5 | 0.40 |
| 2 | DBN-GRU | 98.7 | 98.5 | 98.9 | 98.7 | 1.30 |
| 3 | NMLA-AMDCEF | 94.4 | 93.8 | 94.1 | 93.9 | 5.6 |

The proposed hybrid SVM-MLP model achieves the highest accuracy of 99.60%, outperforming DBN-GRU (98.7%) and NMLA-AMDCEF (94.4%), owing to the complementary strengths of margin-based SVM classification and non-linear MLP learning. It also records the best precision (99.45%) and F-measure (99.5%), reflecting fewer false positives and a well-balanced classification performance. Although its recall (98.87%) is marginally lower than DBN-GRU (98.9%), the difference is negligible and is compensated by superior overall accuracy and precision. Most notably, the proposed model attains the lowest classification error of just 0.40%, significantly better than DBN-GRU (1.30%) and NMLA-AMDCEF (5.6%). These results collectively demonstrate that the hybrid SVM-MLP approach provides a more accurate and reliable solution for Android malware detection than existing methods.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

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

## IV. CONCLUSION

This paper proposed a hybrid machine learning framework combining Support Vector Machine (SVM) and Multi-Layer Perceptron (MLP) for the effective prediction and detection of Android malware, addressing the growing limitations of traditional signature-based detection methods against increasingly sophisticated and obfuscated threats. The proposed methodology followed a systematic pipeline involving dataset selection, preprocessing through data cleaning and normalization, dimensionality reduction using Principal Component Analysis (PCA), and classification through a weighted ensemble of SVM and MLP, thereby leveraging the margin-based robustness of SVM alongside the non-linear pattern-learning capability of MLP. The entire simulation was implemented in the Spyder IDE using Python 3.7, utilizing its extensive machine learning libraries for model development and evaluation. Experimental results demonstrated that the proposed hybrid SVM-MLP model significantly outperformed existing approaches, achieving the highest accuracy of 99.60%, precision of 99.45%, and F-measure of 99.5%, along with the lowest classification error of 0.40%, compared to DBN-GRU and NMLA-AMDCEF. These findings confirm that integrating complementary classifiers within a hybrid architecture enhances detection accuracy and reduces misclassification more effectively than standalone models. Overall, the proposed approach establishes a reliable and robust foundation for Android malware detection, with future work aimed at extending the framework to real-time detection scenarios and larger, more diverse malware datasets.

## REFERENCES

1. Kauser.Sk H, Anu.V M (2025) Hybrid Deep Learning Model for Accurate and Efficient Android Malware Detection Using DBN-GRU. PLoS One 20(5): e0310230. https://doi. org/10.1371/journal.pone.0310230
2. Y. Kim, J. Seo, and H. Choi, “Lightweight Android malware detection using hybrid machine learning for real-time protection,” IEEE Access, vol. 12, pp. 145320–145332, 2024.
3. H. Alamro, W. Mtouaa, S. Aljameel, A. S. Salama, M. A. Hamza and A. Y. Othman, "Automated Android Malware Detection Using Optimal Ensemble Learning Approach for Cybersecurity," in IEEE Access, vol. 11, pp. 72509-72517, 2023, doi: 10.1109/ACCESS.2023.3294263.
4. H. Zhu, Y. Li, R. Li, J. Li, Z. You and H. Song, "SEDMDroid: An Enhanced Stacking Ensemble Framework for Android Malware Detection," in IEEE Transactions on Network Science and Engineering, vol. 8, no. 2, pp. 984-994, 1 April-June 2022, doi: 10.1109/TNSE.2020.2996379.
5. L. Gong, Z. Li, H. Wang, H. Lin, X. Ma and Y. Liu, "Overlay-based Android Malware Detection at Market Scales: Systematically Adapting to the New Technological Landscape," in IEEE Transactions on Mobile Computing, doi: 10.1109/TMC.2021.3079433.
6. I. Almomani et al., "Android Ransomware Detection Based on a Hybrid Evolutionary Approach in the Context of Highly Imbalanced Data," in IEEE Access, vol. 9, pp. 57674-57691, 2021, doi: 10.1109/ACCESS.2021.3071450.
7. F. Mercaldo and A. Santone, "Formal Equivalence Checking for Mobile Malware Detection and Family Classification," in IEEE Transactions on Software Engineering, doi: 10.1109/TSE.2021.3067061.
8. L. N. Vu and S. Jung, "AdMat: A CNN-on-Matrix Approach to Android Malware Detection and Classification," in IEEE Access, vol. 9, pp. 39680-39694, 2021, doi: 10.1109/ACCESS.2021.3063748.
9. L. Gong et al., "Systematically Landing Machine Learning onto Market-Scale Mobile Malware Detection," in IEEE Transactions on Parallel and Distributed Systems, vol. 32, no. 7, pp. 1615-1628, 1 July 2021, doi: 10.1109/TPDS.2020.3046092.
10. W. Yuan, Y. Jiang, H. Li and M. Cai, "A Lightweight On-Device Detection Method for Android Malware," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 51, no. 9, pp. 5600-5611, Sept. 2021, doi: 10.1109/TSMC.2019.2958382.
11. K. Liu, S. Xu, G. Xu, M. Zhang, D. Sun and H. Liu, "A Review of Android Malware Detection Approaches Based on Machine Learning," in IEEE Access, vol. 8, pp. 124579-124607, 2020, doi: 10.1109/ACCESS.2020.3006143.
12. D. Li and Q. Li, "Adversarial Deep Ensemble: Evasion Attacks and Defenses for Malware Detection," in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 3886-3900, 2020, doi: 10.1109/TIFS.2020.3003571.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

## IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details