Journal of Global Research in Electronics and Communication
Volume 2, No. 01, January 2026
Available Online at: www.jgrec.info
RESEARCH PAPER

Machine Learning Approaches for Threat Detection in Android Mobile Networks

Mr. Sachin Manekar
Assistant Professor
Department of Computer Sciences and Applications
Mandsaur University, Mandsaur
Sachin.manekar@meu.edu.in

***Abstract*—The development of mobile technologies and the high number of applications based on Android have considerably increased the number of targets of attack by cybercriminals. Android, which is the most popular mobile platform, is becoming the target of more and more malware and adware that aim to steal the privacy, financial data, and integrity of the system of its users. The paper suggests a complex Android malware detection system based on an Artificial Neural Network (ANN) that can be used to improve the classification of malicious and innocuous applications. The study is based on a sample of 4,464 Android instances and includes necessary preprocessing steps, including handling missing values and outliers, min-max normalization, and feature selection using PCA. Random under-sampling was used to overcome the imbalance in classes prior to training of ANN model. The offered ANN has better results, with the accuracy of 99.38%, precision of 99.24%, recall of 99.42%, and F1-score of 99.61%. Accuracy loss curves, confusion matrix, and ROC analysis evaluation have verified the high generalization of the model, learning stability, and good discriminative properties. Comparative findings also indicate that the ANN performs more successfully than traditional models which include, LSTM (93.9%), SVM (94%) and CNN (97.8%). All in all, the research offers a viable deep learning-driven source of strong Android threat detection, which enhances the safety and well-being of the mobile context.**

Keywords—Android, Malware, Network, Cyber Security, Android Security, Machine Learning, Deep Learning,, Cyber-Attacks, Vulnerabilities.

I. INTRODUCTION

Cyberspace has grown due to the widespread development and use of mobile applications and the Internet. With an emphasis on the performance and security of the Internet of Things (henceforth IoT), the vast array of smart sensors and gadgets that surround us is a significant aspect of lives[1]. The Internet of Things (IoT) is a kind of network that uses communication protocols to link anything to other objects[2][3][4][5]. These days, mobile applications provide consumer services like e-commerce, e-banking, remote access control, and e-wallet, as well as corporate services like border control, forensics, criminal detection, and e-passports [6]. However, mobile platforms and gadgets are current targets since they are under assault and may be insecure, posing security risks[7][8]. It may obtain a person's identification, health information, credit card numbers and passwords, address, phone number, personal images and videos, and a login and password for social media sites like Facebook, Twitter, and LinkedIn by using mobile devices [9][10][11]. The goal of mobile device attacks is to obtain this information. This type of private user information might be utilized for a variety of illicit purposes, such fabricating an identity[12][13][14].

Specifically, the Android operating system is controlling the global mobile market, and Android-based devices have become the main focus of the contemporary cyber-attacks. Android ecosystem openness and the use of third-party apps considerably raise threats of malware and adware attacks on the Android mobile networks[15]. These threats are dynamic and unprecedented hence requiring more sophisticated machine learning methods that are specifically tailored to detect and categorize malicious practices in Android platforms.

It is at this point that the classification of programs becomes important such as malware, adware, and benign with these cybersecurity threats[16][17][18]. This classification procedure, which is often enabled by state-of-the-art machine learning algorithms, enables the identification and mitigation of potential risks as a first line of protection[19][20][21]. By understanding the nuances of this ecosystem, recognizing the cybersecurity challenges, investigating the financial objectives of the adware[22], and emphasizing the need of categorization, they can fortify the mobile ecosystem against the detrimental consequences of criminal activity. sophisticated machine learning algorithms that help improve classification robustness and accuracy by accurately classifying mobile applications as benign, malicious, or adware [23]. Building machine learning models that classify mobile applications into categories like malware, adware, and benign helps accomplish the classification process.

A. Motivation and Contribution

The fast growth of Android-based mobile networks and their extensive use to conduct personal, commercial, and financial transactions has predisposed them to more advanced cyber-attacks. Classic signature-based security controls are no longer adequate to handle the dynamic nature of Android malware, adware, and other malicious applications that often take advantage of the platform openness and third-party application economy. With attackers ever devising sophisticated and evolving methods, intelligent, automated and scalable threat-detection mechanisms are urgently needed. Complex application behavioral patterns can be analyzed with the help of machine learning, which enables accurate identification of both benign and malicious software. Therefore, it is crucial to apply machine learning principles to Android threat detection to improve mobile security, avoid loss of confidential data of users, and safer interactions in the existing digitally connected world. The study contributes to Android Mobile Networks in a number of ways:

- The research comes up with an ANN model that performs highly with a high level of 99% in all test metrics. It has high levels of detection of Android malware with minimum errors.
- It is based on a combination of Python models with WEKA preprocessing to develop an effective and stable workflow. Such a hybrid implementation is more effective in improving the data quality, and in general, the model performance.
- There are also clear accuracy and loss plots in the work to demonstrate the learning behavior of the model. Such visual understandings are facilitated in the interpretation of convergence and training stability.
- The research has a replicable model of mobile threat detection in cybersecurity. It can be adapted for real-world applications and further advanced by future research.

B. *Justification and Novelty*

Novelty of this research is that a highly optimized ANN model has been developed to perform Android malware detection with the benefit of using advanced preprocessing methods namely PCA-based feature selection, normalization, and data balancing. This solution has been explained by the fact that the model is highly accurate and it is even more accurate than LSTM, SVM and CNN and it is also more effective in recognizing intricate malicious patterns. Moreover, its excellent generalization, low errors, and almost perfect ROC-AUC can be regarded as a strength of the tool and its usefulness in real-life Android security applications.

II. Literature Review

The development of this work has been guided and strengthened by several important research projects on Threat Detection in Android Mobile Networks.

Bharathi and Tejaswi (2025) presented work, DL-SecureNet, a deep learning transformer-based approach is introduced that aim at identifying cybersecurity threats in 5G networks. Based on the CIC-DDoS dataset, the model was developed and tested with different 5G attributes such as packet size, packet delay, protocol, and transfer rate. Evaluation of the model produced satisfactory results and granted an overall accuracy of 98.4% with precision of 98.5%, recall 98.3%, and F1 score of 98.4% concerning numerous types of attacks such as DDoS, Botnet, SQL Injection, and Malware Communication [24].

Mahnot, George and Alapatt (2025) explore the application of Bidirectional Recurrent Neural Networks and Long Short-Term Memory networks, which are trained on Traffic data records from NSL-KDD, a widely recognized benchmark dataset. It's a secondary dataset. Experimental results validate the achievement of a cross-validation accuracy of 93.40%, F1 is 91.62% and precision is 90.42%, which is greater than the individual models, such as CNN, BiRNN, or LSTM [25].

Albashayreh et al. (2024) creates a broader security threat concern, such as denial-of-service (DoS) attacks, which can disrupt network functionality. The complexity and decentralization of 5G networks create new vulnerabilities for adversaries, necessitating comprehensive security procedures to identify, mitigate, and prevent DoS attacks in 5G networks. The results revealed that the random forest model demonstrated superior recall of 99.98, while BiLSTM demonstrated exceptional performance with a recall of 98.02 [26].

Tossou and Kacem (2023) provide a method to address the security issues with Android smartphones that mixes deep learning and dynamic analysis. In contrast to other deep learning models like Recurrent Neural Networks (RNN), Radial Basis Function Networks (RBFN), and Self-Organizing Maps (SOM), as well as machine learning models like Random Forest (RF), Support Machine Vector (SVM), Decision Tree, and Naïve Bayes, the DrebinDNN model achieved a record high accuracy rate of 99.20% by using deep learning algorithms on the well-known Drebin dataset [27].

Abdel-Basset, Hawash and Sallam, (2022) provides a new federated deep learning (DL) model (Fed-TH) that collects the temporal and spatial representations of network data in order to hunt cyber threats against ICPSs. The simulation results from two public benchmarks confirm these methods' efficacy in terms of accuracy (92.97%, 92.84%) and f1-scores (91.61%, 90.49%) [28].

Alkahtani and Aldhyani (2022) tested with two Android mobile benchmark datasets. The SVM method used the CICAndMal2017 dataset to get the maximum accuracy (100%). Additionally, the LSTM model used the Drebin dataset to reach a high percentage accuracy of 99.40%. Using the CICAndMal2017 dataset, the SVM method's correlation coefficient was R2 = 100%, whereas LSTM's correlation coefficient was R2 = 97.39% in the Drebin dataset. The SVM, LSTM, and CNN-LSTM algorithms are highly effective in detecting malware in the Android environment, according to results compared with current security solutions [29].

Table I presents an overview of recent Threat Detection studies, presenting cutting-edge models, datasets, significant discoveries, and difficulties encountered.

TABLE I. Comparative Review of Cybersecurity Threat Detection Models

Author	Proposed Work	Dataset	Key Findings	Challenges / Recommendations
Bharathi & Tejaswi (2025)	DL-SecureNet: Transformer-based deep learning model for identifying cybersecurity threats in 5G networks	CIC-DDoS dataset	Achieved 98.4% accuracy, 98.5% precision, 98.3% recall, 98.4% F1-score for multiple attacks such as DDoS, Botnet, SQL Injection, Malware Communication	Further study required to address real-time scalability and deployment challenges in live 5G networks
Mahnot, George & Alapatt (2025)	Application of BiRNN and LSTM for threat detection	NSL-KDD (secondary dataset)	Cross-validation accuracy 93.40%, F1 91.62%, precision 90.42% outperforming CNN, BiRNN, or LSTM individually	Need for evaluation on modern real-world datasets; NSL-KDD considered outdated
Albashayreh et al. (2024)	Threat detection for DoS attacks in 5G networks using machine learning	5G traffic dataset (DoS-focused)	Random Forest achieved recall 99.98%; BiLSTM achieved recall 98.02%	Recommend developing advanced models to address decentralization and emerging 5G vulnerabilities
Tossou & Kacem (2023)	Dynamic analysis combined with deep learning for Android malware detection (Drebin DNN)	Drebin dataset	Achieved 99.20% accuracy, surpassing RNN, RBFN, SOM, RF, SVM, DT, NB	More evaluation needed on large-scale, real-world Android apps to ensure robustness

Abdel-Basset, Hawash & Sallam (2022)	Federated DL model (Fed-TH) for cyber-threat hunting in ICPSs	Two public benchmark datasets	Achieved accuracy 92.97%, 92.84% and F1-scores 91.61%, 90.49%	Requires validation in distributed real-time ICPS deployments
Alkahtani & Aldhyani (2022)	Performance evaluation of SVM, LSTM, CNN-LSTM for Android malware detection	CICAndMal2017, Drebin	SVM achieved 100% accuracy (CICAndMal2017); LSTM 99.40% accuracy (Drebin); R² for SVM = 100%, LSTM = 97.39%	Recommend integrating hybrid ML-DL methods and testing across diverse Android environments

III. Research Methodology

The current research would like to create a valid and strong detection of Android malware through an Artificial Neural Network (ANN), and the purpose of this study is to enhance the security of Android by positively impacting the identification of malicious applications and benign applications. The overall workflow of the process is depicted in Figure 1.

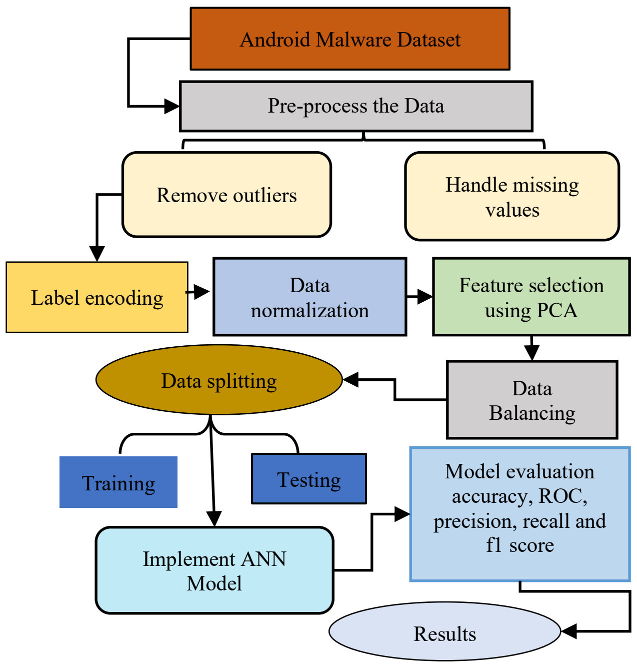

Fig. 1. Proposed flowchart for Threat Detection in Android Mobile Networks

A. Data Collection

The Android Malware Dataset provided the information used in this investigation. The dataset has 4464 balanced samples of Android applications that have been classified as either benign or malicious: Benign = 43.26% and malicious = 56.74%. Attack distribution, feature correlations, and other data visualizations were examined using bar plots, which are provided below:

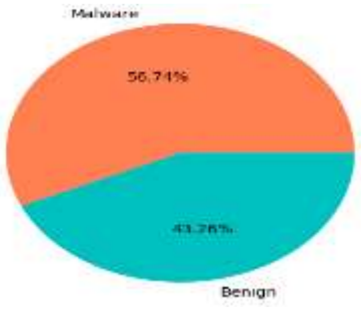

Fig. 2. Category distribution

Figure 2 show pie chart illustrates the distribution between Malware and Benign samples in the dataset. Malware instances account for the larger proportion at 56.74%, while Benign samples make up 43.26%. This indicates that the dataset is slightly imbalanced, with a higher prevalence of malicious samples compared to normal ones. Such a distribution is important to consider during model training, as it may influence the detection accuracy and require balancing techniques to prevent bias toward the majority class.

B. Data Pre-processing

Data preparation involved the process of collecting Android Malware Dataset, merging and cleaning the data, and extraction significant features. The data was then pre-processed to deal with missing values and remove outliers. After this, data transformation and normalization were carried out. Below are the major preprocessing steps:

- **Handle missing values:** Handling missing values is an important process in the data preprocessing to guarantee the quality and reliability of a dataset to be analyzed or trained the machine learning. The methodology selected is based on the type of missing data and the context of the issue.
- **Remove Outliers:** The process of eliminating outliers is achieved by detecting and deleting or adjusting the data points that do not conform in a major way to the rest of the data. This is usually done to enhance the precision and reliability of statistical analyses and ML models.

C. Label Encoding

A label encoder then changes the labels into numerical forms. Label encoding assists in converting particular categorical labels into a form that can be served to an ML algorithm to give better predictions.

D. Data Normalization

The normalization of records was done through the use of the min-max method to restrict the values to between 1 and 0. This was done in order to maximize the performance of the classifiers utilized as well as to reduce the impact of outliers. The normalization was done using the following mathematical Equation (1):

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

In which X denotes the original value of the feature, X' is the normalized value, X_{min} is the minimum value of the feature and X_{max} is the maximum value of the same.

E. Feature selection using Principal Component Analysis (PCA)

Feature selection refers to the process of selecting a subset of useful features out of the set of features in a given dataset. Principal Component Analysis (PCA) is mostly a feature extraction method, but can be used to make a selection of features by examining the significance of the original features in regard to their contribution to the principal components. PCA finds the directions of greatest variance in the data (principal components), and by looking at which of the original features are most significant in them, can choose the most relevant ones.

F. Random Under-Sampling Approach for Data Balancing

Data balancing is a machine learning method that can be applied to classification tasks to address dataset imbalance. Random Under-Sampling is a machine learning tool used to fix the imbalance in the classes within the data. It works by minimizing the number of majority instances to give the minority one a more equal distribution.

G. Data Splitting

The dataset was split into a training and a testing part to test the efficiency of the model. In particular, 70% of the data was to be used in training to estimate the model parameters and the remaining 30% was to be used in testing and evaluating the performance of the model.

H. Proposed Artificial Neural Networks (ANNs) Model

The Artificial Neural Networks (ANN) model is a mathematical model of the human brain's structure and operating system, based on the Artificial Neural Networks model, which is used to identify complex patterns and relationships in data. It is composed of interlocked layers of nodes (neurons), which usually constitute an input layer, a hidden layer or layers, and an output layer. The input data are converted in each neuron based on the weighted connections to the neuron and the activation functions, which allow the network to learn data by using training algorithms like back propagation. With a structure of information that can carry out tasks like classification, pattern recognition, and knowledge "acquired from its environment through a learning process" and stored in synapses, the artificial neural network (ANN) is a popular model that draws inspiration from the way the human brain processes and computes information to carry out a particular task or function. The formulation of neuron k output y is displayed in Equation (2):

$$y_k = \varphi(u_k + b_k) \text{ and } u_k = \sum_{j=0}^{m} w_{kj} x_j \tag{2}$$

Where b_k is the bias that affects the input of the activation function φ, and w_{kj} is the synaptic weight from input x_j to k. A multilayer perceptron's computing strength comes from hidden neurons, which are highly connected to one another through synapses. This allows the network to recognize patterns, which is essential for resolving complicated issues.

Iterative methods based on an error function may be used to train weights and biases. For the n-the iteration on the neuron i, this error function is defined as Equation (3):

$$e_i = d_i(n) - y_i(n) \tag{3}$$

where y is the neuron's output and d are the intended output. This error function may be further extended as an error energy function for an output layer of size $o$in Equation (4):

$$E_{avg} = \frac{1}{N} \sum_{n=1}^{N} E(n) \tag{4}$$

where $E(n) = \frac{1}{2} \sum_{i=1}^{o} e_i^2(n)$

N is the data set's size. The goal of optimization is to decrease the error energy function as much as feasible as it propagates backward through the network, layer by layer.

I. Evaluation Metrics

The proposed system was tested according to a set of evaluation metrics, which include a set of classification evaluation metrics. The fundamental confusion matrix components—True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN)—were obtained by methodically comparing the model predictions with the ground-truth labels. These statistical outcomes were presented to compute the accuracy, precision, recall, and F1-score, each offering a distinct method of the analysis of classification results. This set of metrics is very powerful and multidimensional as the power of prediction of the model and the performance of generalization are considered:

Accuracy: The proportion of the quantity of circumstances accurately estimated by the trained model to the total number of instances in the dataset (input samples). It is given in Equation (5):

$$Accuracy = \frac{\text{TP+TN}}{\text{TP+Fp+TN+FN}} \tag{5}$$

Precision: The precision of a model is defined as the ratio of correctly predicted positive cases to all positive instances. Accuracy shows. The classifier's ability to predict positive classifications is represented by Equation (6):

$$Precision = \frac{\text{TP}}{\text{TP+FP}} \tag{6}$$

Recall: This measure is the proportion of correctly anticipated positive events to all instances that should have turned out to be positive. It is expressed mathematically as Equation (7)-

$$Recall = \frac{\text{TP}}{TP+FN} \tag{7}$$

F1 score: It helps to balance memory and precision because it is a mix of the harmonic mean of both. It has a range of [0, 1]. In mathematics, it is expressed as Equation (8)-

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{8}$$

Receiver Operating Characteristic Curve (ROC): For various decision cut-off points, the ROC is a graphic representation that shows the percentage of cases that are properly identified as positive vs the percentage that are wrongly labeled as positive. FPR is equivalent to 1-specificity, but TPR is sometimes known as sensitivity or recall.

IV. RESULTS AND DISCUSSION

The experimental design entailed the full application of the proposed model in Python in a Jupyter Notebook on a Windows 10 machine with an Intel Core i7 processor and 32 GB RAM. The ANN was trained and tested with Android Malware Dataset and measured on the basis of accuracy (99.38%), precision (99.24%), recall (99.42%), and F1-score (99.61%) shown in Table II, where the model achieved 99% in all of the metrics, and it is capable of detecting malware with a minimal number of errors, which proves its strength in securing Android environments.

TABLE II. EXPERIMENT RESULTS OF PROPOSED MODELS USING ANDROID MALWARE DATASET

Performance Matrix	Artificial Neural Networks (ANNs) Model
Accuracy	99.38
Precision	99.24
Recall	99.42
F1-score	99.61

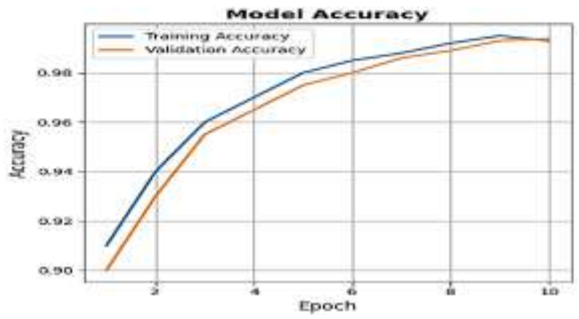

Fig. 3. ANN Model Accuracy graph

The Model Accuracy graph shows the improvement of training and validation accuracy at 10 epochs, represented in Figure 3. The trends of both curves are the same, and the curves are upwards, signifying that the underlying patterns in the data set may be learned by the model. The training and validation accuracy lines are also highly congruent which proves good generalization and constant performance without much overfitting.

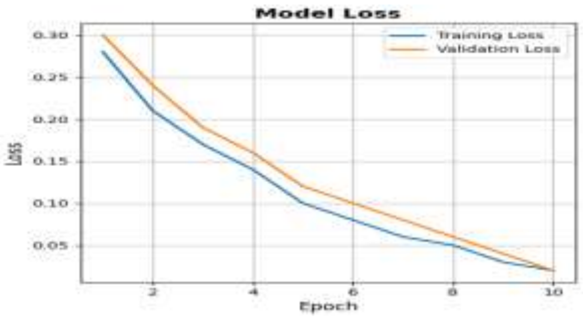

Fig. 4. ANN Model loss graph

The Model Loss graph is used to display the training, as well as the validation loss curve over 10 epochs, as illustrated in Figure 4. Both loss curves decrease gradually, which is an indication of better model optimization with every epoch. The fact that the training and validation loss curves are almost parallel also suggests that the model is balanced in the learning process and is neither underfitted nor overfitted.

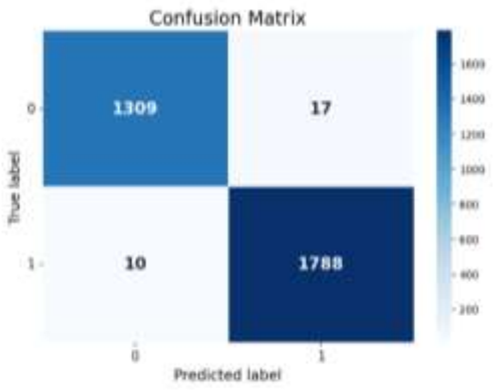

Fig. 5. Confusion matrix for ANN

The confusion matrix provides a thorough explanation of how well the model performed in identifying the two groups shown in Figure 5. It indicates that 1,309 samples each in class 0 and 1,788 samples each in class 1 were correctly classified which implies excellent predictive performance. There were very few misclassifications 17 false positives and 10 false negatives proving that the model has high reliability and the errors of classification are minimal.

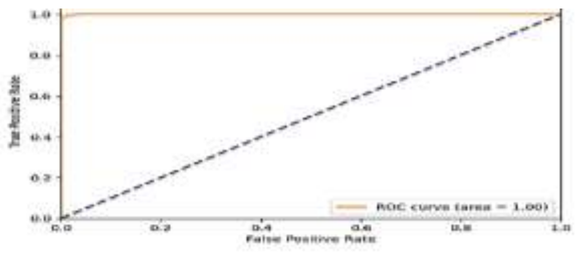

Fig. 6. ROC Curve

The ROC curve shows that the model can determine the two classes in different threshold settings, as represented in Figure 6. The curve remains near the upper left corner, which represents a high discriminative power. With an area under the curve (AUC) of 1.00, the model works nearly flawlessly, demonstrating its strong ability to balance true positive and false positive rates.

A. Comparative Analysis

According to the comparison of accuracy as indicated in Table III, the proposed ANN model has a better performance over the rest of the predictive models that are applied in Android malware threat detection. Although LSTM has 93.9% accuracy, SVM has 94% and CNN has a lot better 97.8, the ANN model has the highest accuracy of 99.38. This comparison shows clearly that ANN model is better in acquiring complex malware patterns, and is therefore the most reliable model among the tested models.

TABLE III. ACCURACY COMPARISON OF DIFFERENT PREDICTIVE MODELS OF THREAT DETECTION USING ANDROID MALWARE DATASET

Models	Accuracy
LSTM [30]	93.9
SVM[31]	94
CNN[32]	97.8
ANN	99.38

The proposed ANN model demonstrates a clear advantage in Android threat detection by achieving the highest accuracy, surpassing other models in performance. This is because it is able to successfully model the complex, non-linear relationships within the Android Malware Dataset and therefore identify subtle patterns of malicious behavior that traditional machine learning models might miss. The ANN has a deep learning architecture that provides a high level of adaptability and robustness to the changing malware, thereby guaranteeing a high rate of reliability and accurate detection of threats, which is of great benefit in ensuring the safety of the Android mobile networks.

V. CONCLUSION AND FUTURE STUDY

The high rate of mobile technological advancement has compounded the issue of security, particularly in the Android operating system, which has an open platform that relies on third-party applications, making it susceptible to subtle malware-related attacks. These were the issues that were taken into account by the author of this paper hence, a model was developed based on Artificial Neural Networks (ANN) to detect malware, with a high accuracy of 99.38%, high precision, recall, and F1-score and thus be able to identify malicious and benign Android apps. High-quality and reliable result was also attained by the model in the classification through systematic preprocessing, normalization, feature selection as well as balancing practices in the Android Malware Dataset that was effective in representing the complex pattern of behavior in the dataset. These results demonstrate that ANN has the potential to approximate non-linear relations and has a substantial threat detection potential in a mobile setting. Although these encouraging results have been achieved, further research can be done to enhance the system by increasing the sample size of real-world and zero-day malware so that generalization is enhanced. The use of more sophisticated deep learning models, including hybrid CNN-RNN models or transformers, can also help to increase detection accuracy. Moreover, one can create lightweight real-time applications on-device detection systems and render AI explainable to bring the applicability, transparency, and scalability of the Android security solutions. Such developments will culminate in a more intelligent and strong malware detection system.

REFERENCES

[1] V. Shah, "A Systematic Review of Formal Methods for Reliable Network Testing and Verification," *NTERNATIONAL Res. J.*, vol. 8, no. 9, pp. 13–19, 2021.

[2] G. Sarraf, "DeepDefender : High-Precision Network Threat Classification Using Adversarial-Resistant Neural Networks," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 2, no. 1, pp. 596–606, 2022, doi: 10.48175/IJARSCT-3600E.

[3] K. Shaukat, S. Luo, V. Varadharajan, I. A. Hameed, and M. Xu, "A Survey on Machine Learning Techniques for Cyber Security in the Last Decade," *IEEE Access*, 2020, doi: 10.1109/ACCESS.2020.3041951.

[4] V. Panchal, "Thermal and Power Management Challenges in High-Performance Mobile Processors," *Int. J. Innov. Res. Sci. Eng. Technol.*, vol. 13, no. 11, 2024, doi: 10.15680/IJIRSET.2024.1311014.

[5] E. Bout, V. Loscri, and A. Gallais, "How Machine Learning Changes the Nature of Cyberattacks on IoT Networks: A Survey," *IEEE Commun. Surv. Tutorials*, vol. 24, no. 1, pp. 248–279, 2022, doi: 10.1109/COMST.2021.3127267.

[6] A. R. Bilipelli, "AI-Driven Intrusion Detection Systems for Large-Scale Cybersecurity Networks Data Analysis : A Comparative Study," *TIJER – Int. Res. J.*, vol. 11, no. 12, pp. 922–928, 2024.

[7] D. Gragnaniello, C. Sansone, and L. Verdoliva, "Iris liveness detection for mobile devices based on local descriptors," *Pattern Recognit. Lett.*, vol. 57, pp. 81–87, May 2015, doi: 10.1016/j.patrec.2014.10.018.

[8] S. Narang and A. Gogineni, "Zero-Trust Security in Intrusion Detection Networks: An AI-Powered Threat Detection in Cloud Environment," *Int. J. Sci. Res. Mod. Technol.*, vol. 4, no. 5, pp. 60–70, Jun. 2025, doi: 10.38124/ijsrmt.v4i5.542.

[9] G. Sarraf, "Autonomous Ransomware Forensics: Advanced ML Techniques for Attack Attribution and Recovery," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 3, no. 3, pp. 1377–1390, Jul. 2023, doi: 10.48175/IJARSCT-11978W.

[10] G. Sarraf, "Behavioral Analytics for Continuous Insider Threat Detection in Zero-Trust Architectures," *Int. J. Res. Anal. Rev.*, vol. 8, no. 4, pp. 596–602, 2021.

[11] S. Singh, "Enhancing Observability and Reliability in Wireless Networks with Service Mesh Technologies," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 5, no. 1, 2025, doi: 10.48175/568.

[12] H.-H. Kim and M.-J. Choi, "Linux kernel-based feature selection for Android malware detection," in *The 16th Asia-Pacific Network Operations and Management Symposium*, IEEE, Sep. 2014, pp. 1–4. doi: 10.1109/APNOMS.2014.6996540.

[13] V. A. Prisacariu, O. Kahler, D. W. Murray, and I. D. Reid, "Real-Time 3D Tracking and Reconstruction on Mobile Phones," *IEEE Trans. Vis. Comput. Graph.*, vol. 21, no. 5, pp. 557–570, May 2015, doi: 10.1109/TVCG.2014.2355207.

[14] N. M. Yip, R. Forrest, and S. Xian, "Exploring segregation and mobilities: Application of an activity tracking app on mobile phone," *Cities*, vol. 59, pp. 156–163, Nov. 2016, doi: 10.1016/j.cities.2016.02.003.

[15] R. Patel, "Automated Threat Detection and Risk Mitigation for ICS (Industrial Control Systems) Employing Deep Learning in Cybersecurity Defence," *Int. J. Curr. Eng. Technol.*, vol. 13, no. 06, pp. 584–591, 2023, doi: 10.14741/ijcet/v.13.6.11.

[16] V. Prajapati, "Enhancing Threat Intelligence and Cyber Defense through Big Data Analytics: A Review Study," *J. Glob. Res. Math. Arch.*, vol. 12, no. 4, 2025.

[17] N. K. Prajapati, "Federated Learning for Privacy-Preserving Cybersecurity: A Review on Secure Threat Detection," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 5, no. 4, pp. 520–528, Apr. 2025, doi: 10.48175/IJARSCT-25168.

[18] P. Nutalapati, J. R. Vummadi, S. Dodda, and N. Kamuni, "Advancing Network Intrusion Detection: A Comparative Study of Clustering and Classification on NSL-KDD Data," in *2025 International Conference on Data Science and Its Applications (ICoDSA)*, IEEE, Jul. 2025, pp. 880–885. doi: 10.1109/ICoDSA67155.2025.11157595.

[19] Vilas Shewale, "Beyond EDR: Exploring the rise of XDR for unified threat detection and response," *World J. Adv. Eng. Technol. Sci.*, vol. 15, no. 2, pp. 380–386, May 2025, doi: 10.30574/wjaets.2025.15.2.0551.

[20] A. Shabtai, L. Tenenboim-Chekina, D. Mimran, L. Rokach, B. Shapira, and Y. Elovici, "Mobile malware detection through analysis of deviations in application network behavior," *Comput. Secur.*, vol. 43, pp. 1–18, Jun. 2014, doi: 10.1016/j.cose.2014.02.009.

[21] S. K. Chintagunta and S. Amrale, "Enhancing Cloud Database Security Through Intelligent Threat Detection and Risk Mitigation," *Tech. Int. J. Eng. Res.*, vol. 9, no. 10, pp. 49–55, 2022, doi: 10.56975/tijer.v9i10.159996.

[22] V. Verma, "Security Compliance and Risk Management in AI-Driven Financial Transactions," *Int. J. Eng. Sci. Math.*, vol. 12, no. 7, pp. 107–121, 2023.

[23] S. Yilmaz and S. Zavrak, "Adware: A Review," *Int. J. Comput. Sci. Inf. Technol.*, vol. 6, no. 6, pp. 5599–5604, 2015.

[24] P. S. Bharathi and R. Tejaswi, "DL-SecureNet: A Transformer-based Cybersecurity Framework for Threat Detection in 5G and Beyond," in *2025 6th International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, IEEE, Jun. 2025, pp. 1276–1282. doi: 10.1109/ICICV64824.2025.11085631.

[25] S. Mahnot, J. P. George, and B. P. Alapatt, "An Algorithm for Cybersecurity Threats Detection in the Internet of Things using Deep Learning Approach," in *2025 6th International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, IEEE, Jun. 2025, pp. 1464–1471. doi: 10.1109/ICICV64824.2025.11085461.

[26] A. Albashayreh, Y. Tashtoush, A. Aldosary, O. Darwish, and F. Albalas, "Explainable-AI for DoS Attacks Detection in 5G Network Using Deep Learning Models," in *2024 International Conference on Intelligent Computing, Communication, Networking and Services (ICCNS)*, IEEE, Sep. 2024, pp. 166–171. doi: 10.1109/ICCNS62192.2024.10776299.

[27] S. Tossou and T. Kacem, "Mobile Threat Detection System: A Deep Learning Approach," in *2023 13th International Conference on Information Science and Technology (ICIST)*, IEEE, Dec. 2023, pp. 323–332. doi: 10.1109/ICIST59754.2023.10367120.

[28] M. Abdel-Basset, H. Hawash, and K. Sallam, "Federated Threat-Hunting Approach for Microservice-Based Industrial Cyber-Physical System," *IEEE Trans. Ind. Informatics*, vol. 18, no. 3, pp. 1905–1917, Mar. 2022, doi: 10.1109/TII.2021.3091150.

[29] H. Alkahtani and T. H. H. Aldhyani, "Artificial Intelligence Algorithms for Malware Detection in Android-Operated Mobile Devices," *Sensors*, vol. 22, no. 6, Mar. 2022, doi: 10.3390/s22062268.

[30] F. Nawshin, R. Gad, D. Unal, and P. N. Suganthan, "Android malware detection and classification using stacked machine learning," *IET Conf. Proc.*, no. 39, pp. 575–584, Jan. 2024, doi: 10.1049/icp.2024.0546.

[31] S. A. Hashmi, "Malware Detection and Classification on Different Dataset by Hybridization of CNN and Machine Learning," *Int. J. Intell. Syst. Appl. Eng.*, vol. 12, no. 6s, 2024.

[32] S. Millar, N. McLaughlin, J. M. del Rincon, and P. Miller, "Android Malware Detection Using Deep Learning," in *Artificial Intelligence and Cybersecurity*, vol. 14, no. 32, 2023, pp. 209–246. doi: 10.1007/978-3-031-15030-2_10.