Asia Pacific Academy of Science Pte. Ltd.

Journal of Biological Regulators and Homeostatic Agents 2026, 40(3), 73.
https://doi.org/10.65746/jbrha73

Article

A novel YOLO-NAS algorithm for android malware detection using histogram of oriented gradients (HOG) based feature extraction

Shanmuga Priya Gopalakrishnan*, Rama Subra Mani Vanamamalai

Department of Electronics and Communication Engineering, Francis Xavier Engineering College, Tirunelveli 627003, Tamilnadu, India
*** Corresponding author:** Shanmuga Priya Gopalakrishnan, priyagshan@outlook.com

CITATION

Gopalakrishnan SP, Vanamamalai RSM. A novel YOLO-NAS algorithm for android malware detection using histogram of oriented gradients (HOG) based feature extraction. Journal of Biological Regulators and Homeostatic Agents. 2026; 40(3): 73. https://doi.org/10.65746/jbrha73

ARTICLE INFO

Received: 26 January 2026
Revised: 18 April 2026
Accepted: 27 April 2026
Available online: 18 May 2026

Abstract: Smartphones have developed into more than just communication tools in the current digital era, becoming essential to many facets of daily life. Because of its large user base and open-source nature, Android is a dominant mobile operating system. But because of its extensive use, it has become a prominent target for more complex malware attacks. This efficient deep learning-based framework for Android virus detection uses a proposed You Only Look Once based Neural Architecture Search (YOLO-NAS) model in conjunction with feature selection techniques. The system first receives a dataset of Android malware, after which data preparation is carried out to enhance data quality, remove noise, and standardize features. A Histogram of Oriented Gradients (HOG) model is used to identify the most pertinent and discriminative features from the processed dataset in order to decrease dimensionality and increase computing efficiency. The YOLO-NAS model, which classifies malware in a two-class environment as either benign or malicious, is then fed the enhanced feature set. The proposed YOLO-NAS concept aims to improve detection accuracy without compromising robustness or scalability. Performance is assessed using common measures such as F1-Score, Accuracy, Precision, and Recall. The suggested method is appropriate for real-time Android security applications since testing results show that YOLO-NAS and enhanced feature extraction greatly enhance malware detection capabilities.

Keywords: android malware detection; deep learning; YOLO-NAS; histogram of oriented gradients (HOG); accuracy

1. Introduction

By spotting intricate patterns in big datasets, deep learning-based Android malware detection seeks to automatically identify harmful apps. Deep learning models have the ability to directly extract high-level features from raw data, including APK files, permissions, API and system calls, network traffic, and opcode sequences, in contrast to conventional signature-based or rule-based approaches. Due to the rapidly increasing quantity of Android malware, its obfuscation, and the potential protection of enormous volumes of data assets held on Android devices, Android malware detection and categorization is considered a huge data problem [1]. With little explainability, dynamic attacks, and biased datasets. The ability of Large Language Models (LLMs) to reason and infer zero-shot offers a possible substitute. However, there are two primary obstacles to overcome when using LLMs for Android malware detection [2]. Obtaining useful information from function calls and graph structures. In this format, we use a graph convolutional neural network technique to generate graph embeddings of the sensitive FCGs related to node characteristics. that SeGDroid obtains an F-score of 96% for malware family classification on the MalRadar dataset

and an F-score of 98% for malware detection on the CICMal2020 dataset. Additionally, the dangerous activities of the Android virus can be tracked using the model description that has been provided [3]. MAPAS uses convolution neural networks (CNN) to assess the behavior of dangerous programs based on their API call graphs. However, MAPAS only employs CNN to find common patterns in malware's API call graphs [4]; it does not use a CNN-generated classifier model. With real-world Android application sets, the deep learning approach can attain up to 96% accuracy, making it particularly well-suited for Android malware detection [5].

Contribution of this work

The following are the primary phases of the suggested approach for further information:

- To choose the most pertinent and discriminative characteristics, a deep feature extraction process employing HOG is included, lowering dimensionality and enhancing detection effectiveness.
- To improve detection speed and accuracy, the sophisticated YOLO-NAS model is modified for two-class Android malware categorization (malicious vs. benign).
- The integrated architecture shows resilience against malware variants by achieving high performance in terms of Accuracy, Precision, Recall, and F1-Score.

The following is a description of the paper's establishment: The introduction is illustrated in Section 1. The pertinent works are described in Section 2. The experiment's results are presented in Section 4, the suggested methodologies are described in Section 3, and a summary and recommendations for additional research are provided in Section 5.

2. Related works

Mallidi et al. [6] have reported, the Android Malware dataset, which includes 4115 Android permissions, a binary class name, and an extraordinarily high dimensionality, is used to evaluate BOAWFS. The collection comprises 407 malware and 1187 benign program samples from 42 distinct malware families, such as ransomware, adware, scareware, and SMS malware. Abubaker et al. [7] have described, the duplicates were removed and classified as malware and non-malware in order to differentiate between benign and dangerous samples. The permissions of an application are extracted upon installation and execution. A binary matrix of (0, 1) binary values contains the features. Zhang et al. [8] have introduced, an MPDroid significantly reduces the training and inferencing time for downstream tasks by merging a multimodal pre-training framework with downstream tasks based on static properties, hence increasing detection efficiency. Experimental findings show that MPDroid works better than existing detection techniques overall, with an average accuracy of 98.3% and an F1-score of 97.6% at a detection time of less than 7.39 seconds.

Habeeb et al. [9] have suggested, an Artificial neural networks (ANNs) are used to identify Android malware, and their effectiveness is compared to other machine learning techniques. With training and testing accuracies of 0. 99 and an average

accuracy of 0. 99, precision of 0. 99, and recall of 0. 98, the ANN model performed well, even if the F1 score an average of precision and recall was 0. 99. Mahindru et al. [10] have introduced, a system that uses Deep Neural Networks (DNNs) to choose features and classifiers. For this study, 1,20,000 Android apps were experimentally scored using five different feature selection methods. Of these, a collection of features developed by Principal Component Analysis (PCA) can identify 94% of Android malware from real-world apps. Bayazit et al. [11] have proposed, the publicly available CICInvesAndMal2019 dataset using some cutting-edge RNN-based deep learning methods. The recommended models have proficient accuracy and detecting capabilities. With an accuracy percentage of 98.85%, the BiLSTM model outperformed the other suggested methods.

3. Proposed system

A thorough process for detecting Android malware utilizing the YOLO-NAS method is shown in **Figure 1**.

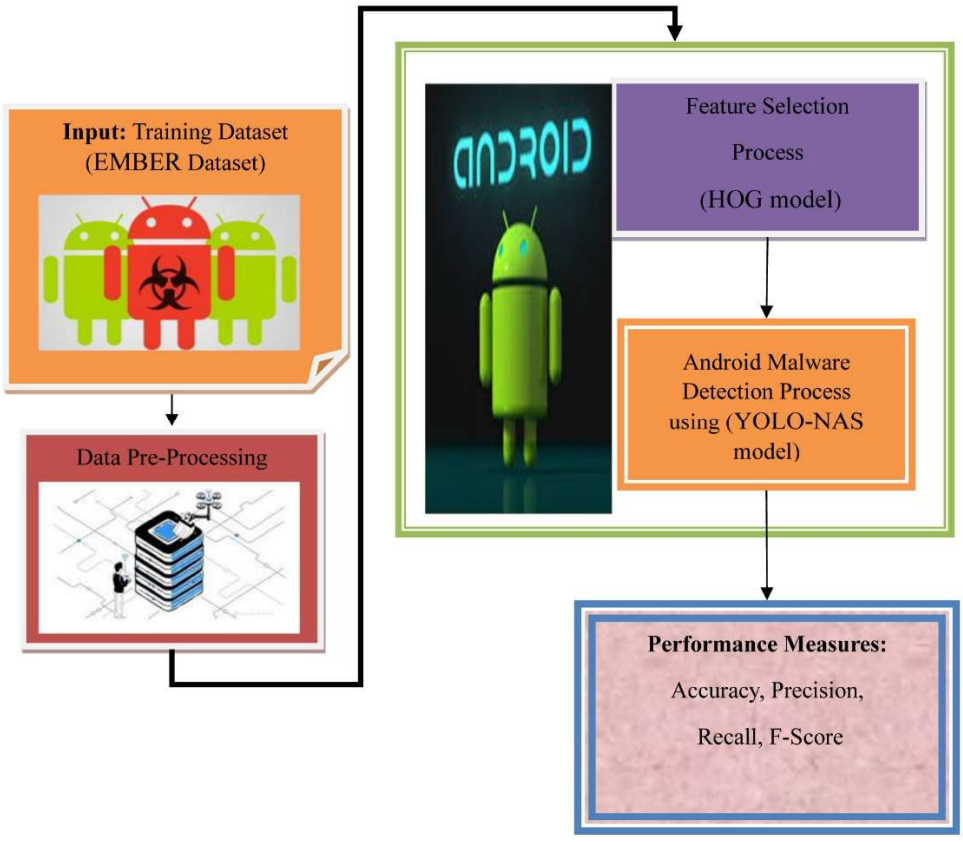

Figure 1. Block diagram for the suggested YOLO-NAS technique to detect Android malware.

An efficient deep learning framework for Android virus detection that blends feature selection techniques with the recommended YOLO-NAS model. The system first receives a dataset of Android malware, after which data preparation is carried out to enhance data quality, remove noise, and standardize features. To reduce dimensionality and improve computation speed, a HOG model for feature selection is

used to extract the most relevant and discriminative features from the processed dataset. The improved feature set is then provided to the YOLO-NAS model, which performs high-performance malware classification in a two-class configuration benign vs. harmful. The proposed architecture aims to improve detection accuracy without compromising robustness or scalability.

3.1. Data pre-processing

Prior to data analysis, raw data must be properly pre-processed. Ensuring the correctness, completeness, and consistency of the data is the primary goal of data pre-processing, which creates a trustworthy database for further analysis. Data normalization, missing value imputation, outlier reduction, and oversampling are all part of the data preprocessing pipeline. The process of transforming data to conform to a particular scale range or a standard normal distribution is known as data standardization. Assigning similar scales to data with different characteristics aims to eliminate the impact of multiple scales. The process of guessing or filling in missing values in a data collection is known as missing value imputation, and it is done to maintain data availability and integrity. By using suitable imputation approaches to infer and fill in missing values, such as mean imputation, regression imputation, or model-based imputation based on the patterns of existing data, the impact of data loss on analytic results can be minimized.

3.2. Feature extraction using HOG model

For finding patterns and textures in pictures, HOG is a very powerful feature extraction technique. The system can identify the texture structure of batik data thanks to the HOG features, which record the orientation and gradient magnitude information included in an image. The following procedures can be used to mathematically explain HOG:

Data Division: A number of tiny grid cells, such as 8x8 pixels, are created from the batik image.

Gradient Calculation: The Sobel operator is used to determine the gradient's magnitude and orientation for each cell, with G_x and G_y standing for the image gradients in the horizontal and vertical directions, respectively.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{1}$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{2}$$

Histogram Formation: Each cell produces a gradient direction histogram that shows how the orientation angles are distributed throughout the image. The following formulas are used to determine the magnitude and orientation.

$$M = |G_x| + |G_y| \tag{3}$$

$$\theta = arctan\left(\frac{G_x}{G_y}\right) \quad (4)$$

HOG Feature Calculation: The features formed from HOG are determined by calculating statistical variables like mean, standard deviation, skewness, and kurtosis after the HOG data has been obtained.

3.3. Proposed YOLO-NAS modal based android malware detection process using algorithm

The YOLO NAS deep learning model algorithm is shown in **Figure 2**.

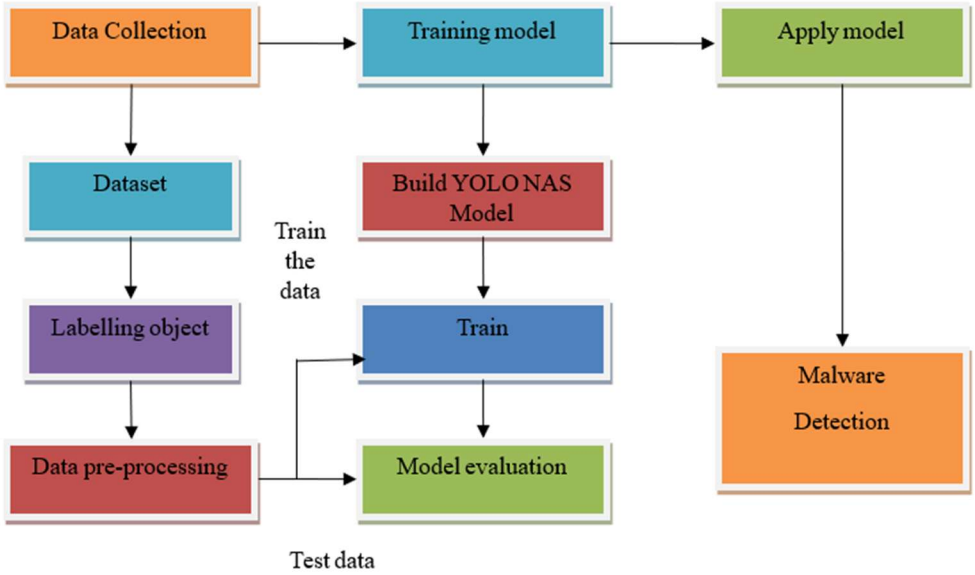

Figure 2. YOLO- NAS deep learning model algorithm.

The process begins with data collection, which involves gathering relevant data samples and organizing them into a dataset. After that, the gathered data is subjected to object labeling, which identifies the key characteristics or classes in order to get them ready for supervised learning. Data preparation is done after labeling to improve training performance by cleaning, normalizing, and improving the data. Training and testing sets are separated from the prepared data. The YOLO-NAS model is then constructed and trained using the processed training data. The model picks up features and patterns from the dataset during training. Test data is used to assess the model after training and calculate performance indicators such as accuracy. Finally, the validated model is applied for malware detection, where it identifies and classifies malicious instances effectively. Algorithm 1 shows the YOLO NAS deep learning model.

3.4. Performance evaluation

The performance of the proposed algorithm is evaluated by calculating the following percentages for sensitivity (SE), specificity (SP), and accuracy (ACC).

$$\text{Accuracy}=\frac{\text{TP}}{\text{TP+TN+FP+FN}} \quad (5)$$

$$\text{Sensitivity}=\frac{\text{TP}}{\text{TP+TN}} \quad (6)$$

$$\text{Precision}=\frac{\text{TP}}{\text{TP+FP}} \quad (7)$$

$$\text{Recall}=\frac{\text{TP}}{\text{TP+FN}} \quad (8)$$

Algorithm 1 YOLO NAS deep learning model

Input: The training data dataset

Output: Noise reduction, YOLO NAS, and attention weight.

First, high-level and low-level features are extracted from training data simultaneously using the YOLO NAS and Efficient Det methods.

The activations for Model = YOLO NAS are:

Mish

more seamless activation procedure

Hard Swish characteristics are extracted.

The fusion station most likely combines feature maps from the equations of both algorithms using element-wise addition.

YOLO NAS employs an attention mechanism equation in the DETR3 transformer.

End for

4. Experimental result and discussion

The EMBER dataset consists of eight categories of raw characteristics, such as format-agnostic histograms, string counts, and parsed features. The malware database was used to evaluate the YOLO-NAS technique's simulated results [12,13].

A confusion matrix for a binary malware detection model that divides samples into malware and benign categories is shown in **Figure 3** and **Table 1** above. The comparison between the expected and real labels is displayed in the matrix. Five of the 500 real malware samples are mistakenly categorized as innocuous, while 495 are appropriately labeled as malware. In the same way, 496 of 500 real benign samples are successfully detected, whereas 4 are mistakenly categorized as malware. The model achieves an overall accuracy of 99.04%, showing strong detection capabilities with very low error rates, with 991 correct predictions out of 1000 samples. This shows how well the suggested malware detection system can differentiate between harmful and benign files.

Table 1. Details about the database.

Classes	Predicted malware	Predicted benign
Actual malware	495	5
Actual benign	4	496

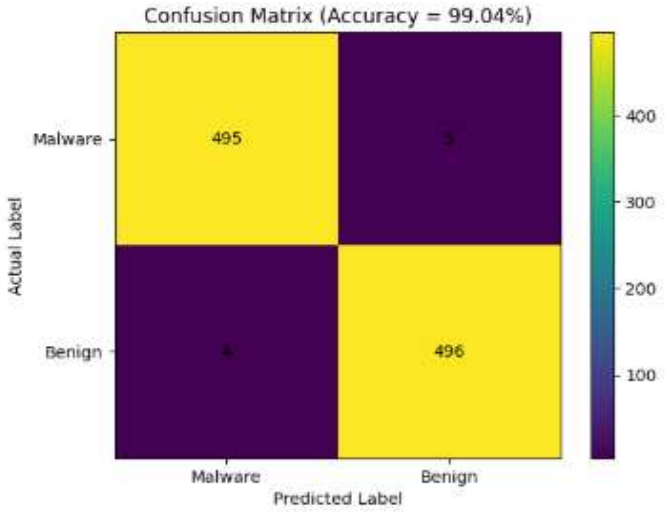

Figure 3. Confusion matrix for YOLO-NAS model-based malware detection.

Figure 4 displays the TRA and TES accuracy curves, which are used to evaluate the efficacy of the YOLO-NAS approach. The TRA and TES accuracy curves grew as the epoch number rose [14].

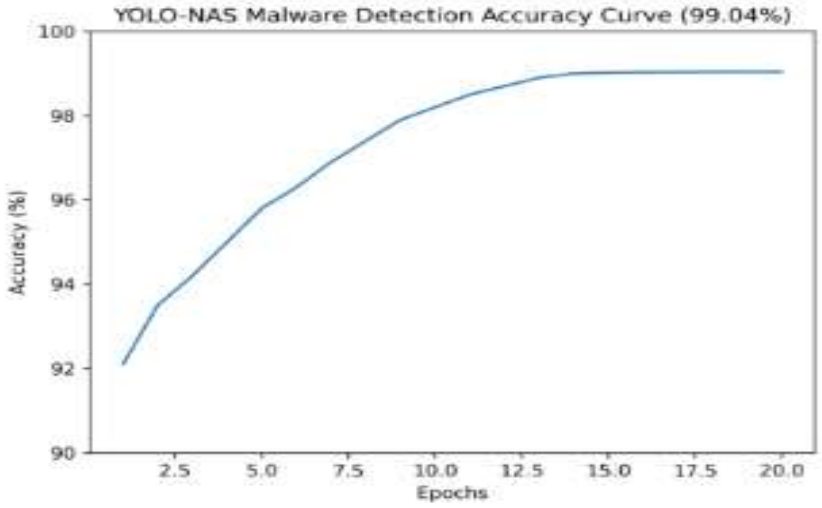

Figure 4. Accuracy curve of YOLO-NAS methodology.

A comparative performance analysis of various deep learning algorithms for malware detection using common assessment metrics including Accuracy, Precision, Recall, and F1-score is shown in **Table 2** and **Figure 5**. With a 90.81% F1-score, 95.19% accuracy, 92.58% precision, and 89.10% recall, the CNN models show respectable but rather weak detection abilities. With an accuracy of 98.21% and a balanced F1-score of 98.12%, the CNN-LSTM hybrid model demonstrates the benefits of integrating spatial and sequential feature learning. The GNN model also performs well, consistently achieving precision and recall values of 97.63%. With 99.04% accuracy, 99.20% precision, 99% recall, and 99.10% F1-score, the suggested YOLO-NAS model outperforms all current techniques and demonstrates exceptional

classification abilities and balanced identification of both benign and harmful applications.

Table 2. Comparative outcome of YOLO-NAS methodology with exising systems.

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
CNN models (MobileNet-V2, ResNet, DenseNet, Inception-V3) [15]	95.19	92.58	89.10	90.81
CNN-LSTM [16]	98.21	97.85	98.40	98.12
GNN [17]	97.63	97.20	97.95	97.57
YOLO-NAS (Proposed)	99.04	99.20	99	99.10

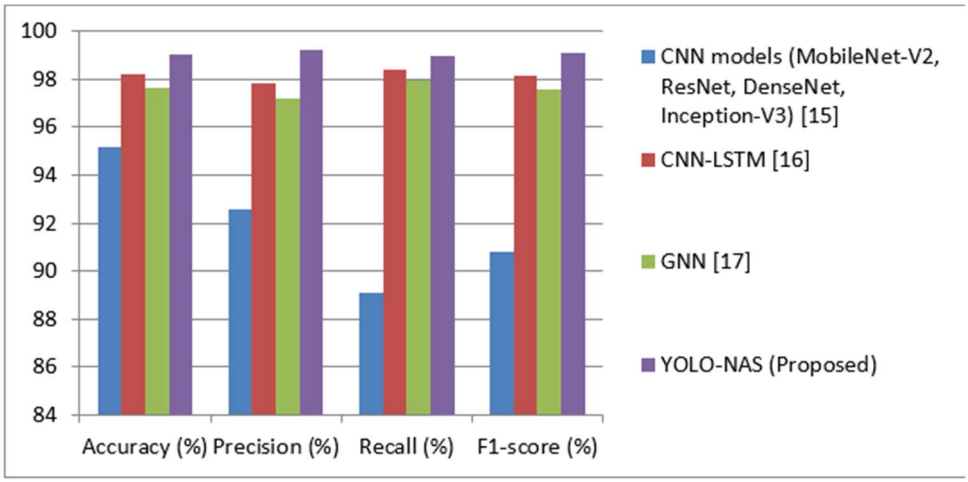

Figure 5. Comparison of the suggested and existing models' performances.

5. Conclusion

High classification performance and robustness are achieved by the proposed Android malware detection system through the efficient use of cutting-edge deep learning algorithms. By employing the HOG model for feature selection, the model successfully removes superfluous and redundant features, improving computing efficiency and discriminative capacity. The improved features are then processed using the YOLO-NAS model, which provides accurate and fast two-class categorization of benign and malicious programs. The suggested GAN-YOLOV7 has an accuracy of 99.04% outperforms conventional malware detection techniques, making it a reliable and effective way to counter evolving Android malware threats. Use more and more varied datasets in the future to enhance generalization and flexibility in response to new malware threats.

Author contributions: Conceptualization, SPG and RSMV; methodology, SPG; software, SPG; validation, SPG and RSMV; formal analysis, SPG; investigation, SPG; resources, RSMV; data curation, RSMV; writing—original draft preparation, RSMV; writing—review and editing, SPG; visualization, RSMV; supervision, RSMV; project administration, SPG; funding acquisition, RSMV. All authors have read and agreed to the published version of the manuscript.

Funding: None.

Ethical approval: Not applicable.

Informed consent statement: Not applicable.

Conflict of interest: The authors declare no conflict of interest.

References

1. Qiu J, Zhang J, Luo W, et al. A survey of android malware detection with deep neural models. ACM Computing Surveys (CSUR). 2020; 53(6): 1–36. doi: 10.1145/3417978
2. Qian X, Zheng X, He Y, et al. Lamd: Context-driven android malware detection and classification with llms. 2025 IEEE Security and Privacy Workshops (SPW). IEEE; 2025. pp. 126–136. doi: 10.1109/SPW67851.2025.00017
3. Liu Z, Wang R, Japkowicz N, et al. SeGDroid: An Android malware detection method based on sensitive function call graph learning. Expert Systems with Applications. 2024; 235: 121125. doi: 10.1016/j.eswa.2023.121125
4. Kim J, Ban Y, Ko E, et al. MAPAS: A practical deep learning-based android malware detection system. International Journal of Information Security. 2022; 21(4): 725–738. doi: 10.1007/s10207-022-00579-6
5. Odusami M, Abayomi-Alli O, Misra S, et al. Android malware detection: A survey. International conference on applied informatics. Cham: Springer International Publishing; 2018. pp. 255–266. doi: 10.1007/978-3-030-01535-0_19
6. Mallidi S. Bat optimization algorithm for wrapper-based feature selection and performance improvement of android malware detection. IET Networks (Wiley-Blackwell). 2021; 10(3). doi: 10.1049/ntw2.12022
7. Abubaker H, Ali A, Shamsuddin SM, Hassan S. Exploring permissions in android applications using ensemble-based extra tree feature selection. Indonesian Journal of Electrical Engineering and Computer Science. 2020; 19(1): 543–552. doi: 10.11591/ijeecs.v19.i1.pp543-552
8. Zhang S, Su H, Liu H, et al. MPDroid: A multimodal pre-training Android malware detection method with static and dynamic features. Computers & Security. 2025; 150: 104262. doi: 10.1016/j.cose.2024.104262
9. Habeeb MA, Khaleel Y. Enhanced android malware detection through artificial neural networks technique. Mesopotamian Journal of CyberSecurity. 2025; 5(1): 62–77. doi: 10.58496/MJCS/2025/005
10. Mahindru A, Sangal AL. Deepdroid: Feature selection approach to detect android malware using deep learning. In 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS). IEEE; 2019. pp. 16–19. doi: 10.1109/ICSESS47205.2019.9040821
11. Çalık Bayazıt E, Şahingöz ÖK, Doğan B. A deep learning based android malware detection system with static analysis; 2022. pp. 1–6.
12. Wasif MS, Miah MP, Hossain MS, et al. "CNN-ViT synergy: An efficient Android malware detection approach through deep learning." Computers and Electrical Engineering. 2025; 123: 110039. doi: 10.1016/j.compeleceng.2024.110039
13. Vinayakumar R, Alaza M, Soman KP, et al. Robust intelligent malware detection using deep learning. IEEE Access. 2019; 7: 46717–46738. doi: 10.1109/ACCESS.2019.2906934
14. Gopinath M, Sethuraman SC. A comprehensive survey on deep learning based malware detection techniques. Computer Science Review. 2023; 47: 100529. doi: 10.1016/j.cosrev.2022.100529
15. Abualhassan Z, Hassan E, Husni D, et al. Malware recognition using novel convolutional neural network with residual connections. International Journal of Machine Learning and Cybernetics. 2026; 17(3): 86. doi: 10.1007/s13042-025-02815-6
16. Musa HO, Muhanad TY. "Effective android malware detection using CNN and LSTM Model with GWO-Based feature selection." Babylonian Journal of Machine Learnin. 2026; (2026): 1–22. doi: 10.58496/BJML/2026/001

17. Lo WW, Layeghy S, Sarhan M, et al. Graph neural network-based android malware classification with jumping knowledge. 2022 IEEE Conference on Dependable and Secure Computing (DSC). IEEE; 2022. pp. 1–9. doi: 10.1109/DSC54232.2022.9888878