

Byte to image mapping: A new paradigm for Malware Detection in internet of Things Devices

Gunji Nagaraju¹, Ch.Pavani²

¹M. Tech Student, ²Assistant Professor, ^{1,2}Department of Electronics and Communications Engineering, ^{1,2}Mother Teresa Institute of Science and Technology, Sathupally, Khammam 507303, Telangana

ABSTRACT

With the rapid expansion of Internet of Things (IoT) devices, the number of connected devices worldwide is projected to exceed 29 billion by 2030, while reports indicate that over 57% of these devices remain vulnerable to malware attacks. This surge in interconnected systems demands robust and scalable detection mechanisms to safeguard against evolving cyber threats. Existing malware detection approaches often suffer from high false positive rates and limited adaptability to the diverse architectures of IoT devices, making them less effective in real-time environments. This work introduces a novel *Byte-to-Image* mapping framework for malware detection, transforming raw binary files into grayscale images for pattern recognition. The proposed methodology begins with comprehensive image preprocessing to normalize and enhance structural features of malware patterns. Traditional machine learning models, including Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), and Support Vector Machine (SVM), are evaluated for baseline performance. To enhance detection accuracy and feature discrimination, we propose a hybrid approach where image features are extracted using the deep learning model VGG19 and then classified using an Extra Trees Classifier (ETC). This combination leverages VGG19's hierarchical feature extraction capability with ETC's robustness in handling high-dimensional data, resulting in a highly accurate and computationally efficient detection system. Experimental results demonstrate that the proposed model

outperforms traditional methods, offering improved precision and recall for both Normal and Malware classifications, making it a promising solution for securing IoT ecosystems.

1. INTRODUCTION

The Internet of Things (IoT) has rapidly evolved into a transformative technology, connecting billions of devices across industries such as healthcare, transportation, manufacturing, and smart homes. According to industry forecasts, the number of IoT devices worldwide is expected to surpass 29 billion by 2030, driven by the integration of sensors, actuators, and embedded systems into everyday objects. This massive interconnection facilitates seamless data exchange and automation, improving operational efficiency and enabling intelligent decision-making. However, the rapid proliferation of IoT systems also significantly increases the digital attack surface, creating new avenues for cybercriminals to exploit. With reports indicating that over 57% of IoT devices remain vulnerable to security breaches, the urgency for robust detection mechanisms has never been greater. Cybersecurity threats targeting IoT devices have become increasingly sophisticated, with malware designed to evade traditional signature-based detection methods. Unlike conventional computing platforms, IoT devices often operate with limited processing power, memory, and energy resources, making it challenging to deploy heavy security solutions. Moreover, the heterogeneity of IoT hardware and operating systems introduces additional complexity, as security protocols must

adapt to a wide range of architectures and communication standards. In this environment, malware infections can lead to severe consequences, such as large-scale botnet formations, unauthorized data exfiltration, service disruptions, and even physical harm in safety-critical systems. High-profile incidents like the Mirai botnet attack have demonstrated the devastating impact such vulnerabilities can have on global networks. Another major challenge lies in the dynamic and evolving nature of IoT malware. Attackers frequently modify malicious code to alter its signature, rendering many conventional detection approaches ineffective. These polymorphic and metamorphic malware strains can bypass static analysis techniques, while encrypted payloads and obfuscation methods make reverse engineering difficult.

2. LITERATURE SURVEY

Tariq et al. [6] proposed a parameter-tuned Deep Learning (DL) model for malware visualization and classification by transforming binary files into grayscale images and optimizing hyperparameters for improved detection accuracy. The approach incorporated Convolutional Neural Networks (CNN) for extracting hierarchical spatial patterns from malware images and applied parameter tuning to enhance classification precision. The methodology addressed class imbalance through preprocessing techniques and utilized visualization to reveal distinctive malware family characteristics. Zhao et al. [7] developed Apppoet, a Large Language Model (LLM)-based Android malware detection framework that employed multi-view prompt engineering to extract semantic and behavioral features from application data. The system analyzed Android application permissions, API calls, and control-flow graphs, integrating multiple feature perspectives for classification. Prompt engineering enabled the model to contextualize features effectively, improving detection in dynamic

and static scenarios. The architecture leveraged the adaptability of LLMs to interpret heterogeneous features for robust classification outcomes. Alam et al. [8] introduced MIRACLE (Malware Image Recognition And Classification by Layered Extraction), a framework that converted malware binaries into RGB images and extracted features through a layered representation approach. The model exploited multi-level feature extraction to capture spatial and spectral patterns, enhancing classification accuracy. Layered extraction improved the separation of malware families by emphasizing discriminative patterns at varying abstraction levels.

Zhang et al. [9] presented IMCMK-CNN, a lightweight CNN architecture incorporating multi-scale kernels for image-based malware classification. The multi-scale kernels allowed the network to capture both local and global patterns from malware images, enhancing feature diversity while maintaining a compact model size. The approach improved efficiency by reducing parameter count without compromising accuracy. The model achieved superior classification results on multiple datasets, proving effective for deployment in resource-constrained environments. Nethala et al. [10] proposed a deep learning-based ensemble framework for Android malware detection, integrating multiple neural network architectures to improve generalization and resilience against evasion techniques. The ensemble combined predictions from different classifiers to achieve higher accuracy and robustness against adversarial manipulation. The methodology applied both static and dynamic analysis features to strengthen detection reliability. Experimental results showed that the ensemble framework outperformed individual models in various test scenarios.

3. Proposed System

The proposed Byte-to-Image Mapping with VGG19 Feature Extraction and Extra Trees Classification method is not presented as shown in Figure 1 in existing surveys and introduces a novel combination that overcomes the limitations of current malware detection approaches for IoT devices. Unlike traditional byte-code analysis or direct machine learning models (such as LDA, QDA, and SVM) that rely on handcrafted features, our method integrates image preprocessing with deep hierarchical feature learning from VGG19, followed by a lightweight but highly discriminative Extra Trees Classifier.

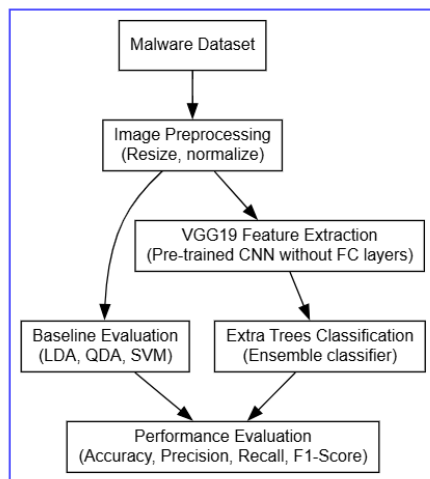


Figure 1: Proposed System Architecture.

4.2 VGG19 Feature Extraction

VGG19 is a deep CNN as shown in Figure 2 known for its simplicity in architecture yet exceptional performance in feature extraction tasks. Its consistent use of small 3×3 convolution filters allows it to capture fine-grained spatial information, making it highly effective in differentiating subtle patterns, textures, and shapes present in images. The depth of 19 layers enables the model to progressively learn hierarchical representations, from low-level edges to high-level abstract features. For malware detection using byte-to-image mapping, VGG19 excels at translating visual patterns into discriminative feature vectors that can significantly boost classification

performance when integrated with machine learning classifiers.

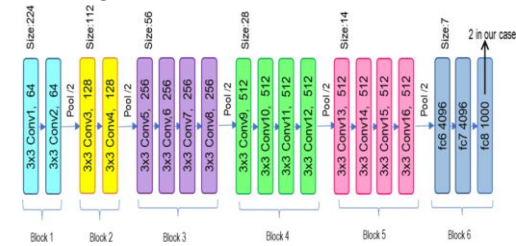


Figure 2: VGG19 Architecture.

4.3 Existing LDA Classifier

Linear Discriminant Analysis (LDA) as shown in Figure 3 is a supervised classification algorithm that not only classifies data but also reduces dimensionality by projecting data into a space that maximizes class separability. It is particularly effective when the dataset follows a Gaussian distribution with equal covariance across classes. LDA is computationally efficient, interpretable, and works well even with relatively small datasets.

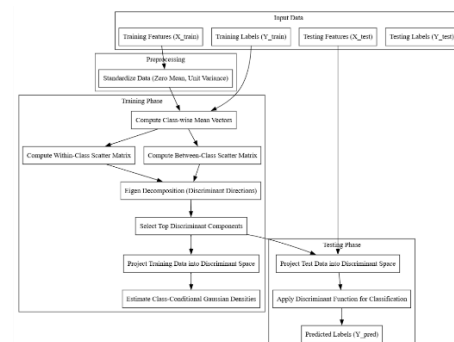


Figure 3: Existing LDA Flowchart.

4.4 Existing QDA Classifier

Quadratic Discriminant Analysis (QDA) as shown in Figure 4 is a probabilistic classifier that assumes each class follows a Gaussian distribution with its own covariance matrix, enabling it to model complex, non-linear decision boundaries.

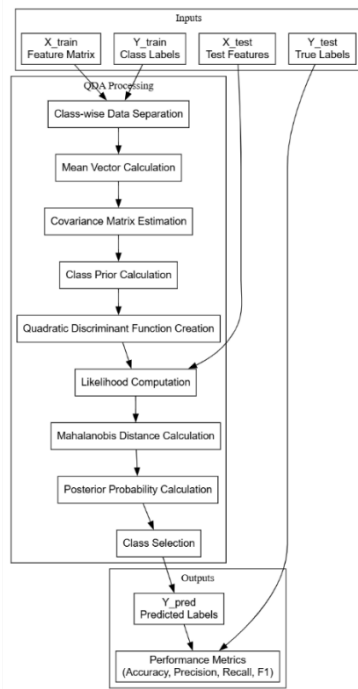


Figure 4: Existing QDA Flowchart.

This flexibility allows QDA to separate classes more effectively when they are not linearly separable. Unlike Linear Discriminant Analysis (LDA), QDA adapts to class-specific spread and orientation, providing more accurate classification in heterogeneous datasets. It works well in scenarios where classes have distinct variances, making it suitable for applications such as image-based malware detection, medical diagnosis, and speech recognition.

4.5 Existing SVM Classifier

Support Vector Machines (SVM) as shown in Figure 4.5 are powerful supervised learning models that excel in high-dimensional feature spaces, making them ideal for tasks where feature extraction has already been performed (e.g., from CNNs like VGG19). SVM works by finding the optimal separating hyperplane that maximizes the margin between different classes, ensuring strong generalization even in cases of small datasets. With kernel functions such as RBF, polynomial, or sigmoid, SVM can model complex non-linear decision boundaries, which is particularly useful when malware signatures in feature space do not follow

simple separations. Additionally, SVM is robust to overfitting when tuned properly and performs well even with noisy or unbalanced datasets.

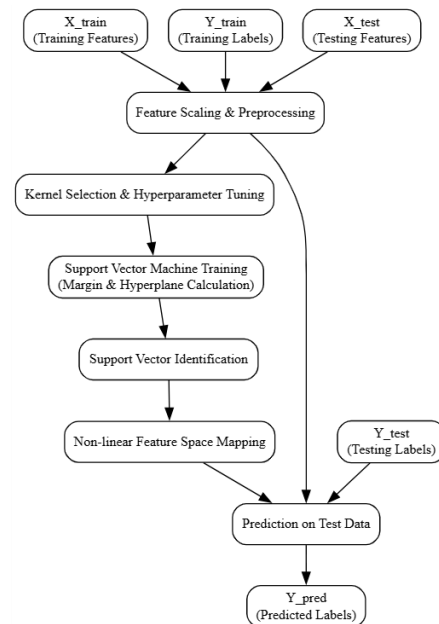


Figure 5: Existing SVM Classifier.

4.6 Proposed Extra Trees Classifier

The Extra Trees Classifier (Extremely Randomized Trees) as shown in Figure 4.6 is a powerful ensemble learning method that builds multiple decision trees using randomized feature selection and random split points. Unlike traditional decision trees, Extra Trees introduces randomness not only in feature selection but also in choosing split thresholds, which reduces overfitting and improves generalization on unseen data. Its computational efficiency is higher compared to some other ensemble methods since it does not rely on bootstrapped sampling and processes the entire dataset directly. For malware detection, the Extra Trees Classifier excels in handling high-dimensional feature spaces generated by methods like VGG19 feature extraction. It captures complex feature interactions while remaining robust to noise and outliers.

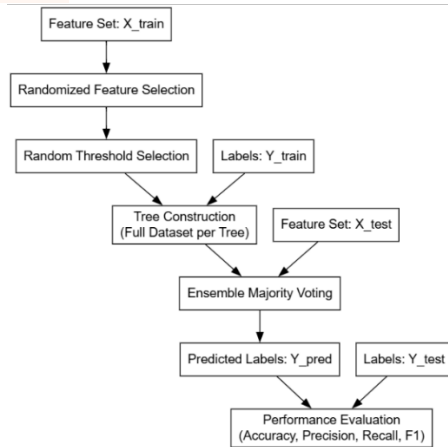


Figure 6: Proposed Extra Trees Classifier.

4. RESULTS AND DISCUSSION

The research work is a Tkinter-based GUI application for malware detection in IoT devices using a byte-to-image mapping paradigm. It leverages deep learning (VGG19 for feature extraction) and machine learning models (LDA, QDA, ETC, SVM) to classify images derived from malware and benign files. The application allows users to upload datasets, preprocess images, train models, evaluate performance, and predict classifications for new images. Figure 7 shows the graphical user interface (GUI) of the malware detection application based on byte-to-image mapping. The interface features a title "Byte to image mapping: A new paradigm for Malware Detection in Internet of Things Devices" at the top, set against a background image with a grid-like pattern. On the left side, there are buttons labeled "Upload Dataset," "VGG19 Feature extraction," "Train Test Splitting," "Existing LDA," "Existing QDA," "Train SVM," "Train ETC," "Upload test image," "Model Comparison Graphs," and "Exit," allowing users to interact with various functionalities.



Figure 7: GUI of Research Work.

Figure 8 shows the Existing LDA ROC Curve, which plots the True Positive Rate (TPR) against the False Positive Rate (FPR) to evaluate the performance of the Linear Discriminant Analysis (LDA) model. The ROC curve is represented by an orange line with an area under the curve (AUC) of 0.97, indicating a high level of accuracy in distinguishing between benign and malware classes. The dashed blue line represents the random guess line (AUC = 0.5), showing that the LDA model performs significantly better than random classification, with the curve closely hugging the top-left corner.

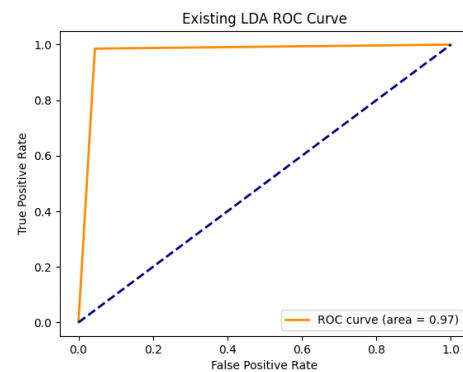


Figure 8: Existing LDA RoC Curve.

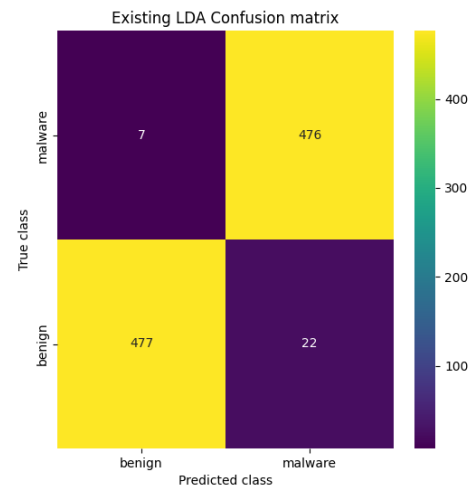


Figure 9: Existing LDA Confusion Matrix.

Figure 8 shows the Existing LDA Confusion Matrix, which summarizes the performance of the LDA model on the test dataset. The matrix is a 2x2 grid with rows labeled "benign" and "malware" (true classes) and columns labeled "benign" and "malware" (predicted classes). The values

are 477 true negatives (benign correctly predicted as benign), 22 false positives (benign predicted as malware), 7 false negatives (malware predicted as benign), and 476 true positives (malware correctly predicted as malware), indicating a strong performance with a high number of correct classifications. Figure 9 shows the Existing QDA ROC Curve, which plots the True Positive Rate against the False Positive Rate for the Quadratic Discriminant Analysis (QDA) model. The orange ROC curve has an AUC of 0.96, suggesting excellent discriminatory power between benign and malware classes. The dashed blue line (AUC = 0.5) serves as a baseline, and the QDA model's curve, while slightly below the LDA's AUC, still indicates robust performance with a steep rise towards the top-left corner.

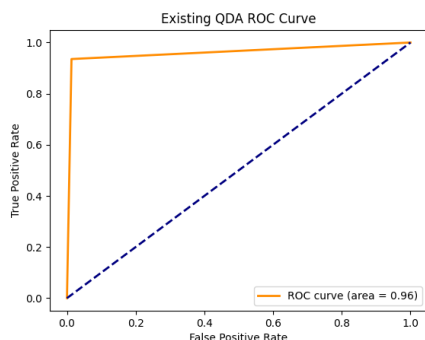


Figure 9: Existing QDA RoC Curve

Figure 10 shows the Existing QDA Confusion Matrix, which details the QDA model's classification results. The 2x2 matrix shows 450 true negatives (benign correctly predicted as benign), 6 false positives (benign predicted as malware), 34 false negatives (malware predicted as benign), and 492 true positives (malware correctly predicted as malware). This distribution reflects a high accuracy, with a slightly better true positive rate compared to LDA, though with a few more false negatives.

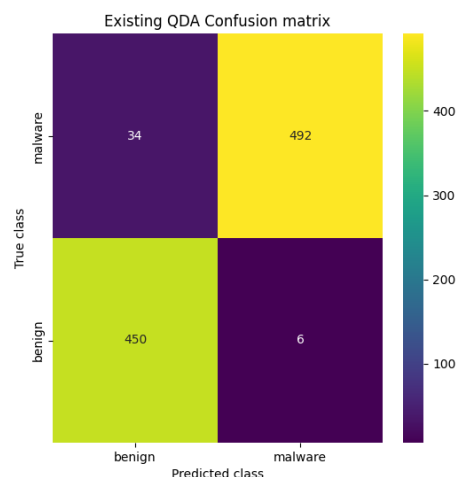


Figure 10: Existing QDA Confusion Matrix.

Figure 11 shows the Tuned SVM Classifier ROC Curve, which illustrates the performance of the Support Vector Machine (SVM) model with tuned hyperparameters. The orange ROC curve has an AUC of 0.96, indicating a strong ability to differentiate between classes. The dashed blue line (AUC = 0.5) highlights that the SVM model significantly outperforms random guessing, with the curve showing a steep initial rise, reflecting good sensitivity and specificity. Figure 12 shows the Tuned SVM Classifier Confusion Matrix, which presents the SVM model's predictions. The 2x2 matrix contains 473 true negatives (benign correctly predicted as benign), 28 false positives (benign predicted as malware), 11 false negatives (malware predicted as benign), and 470 true positives (malware correctly predicted as malware). This shows a balanced performance with a high number of correct predictions, though with a slightly higher false positive rate than LDA and QDA.

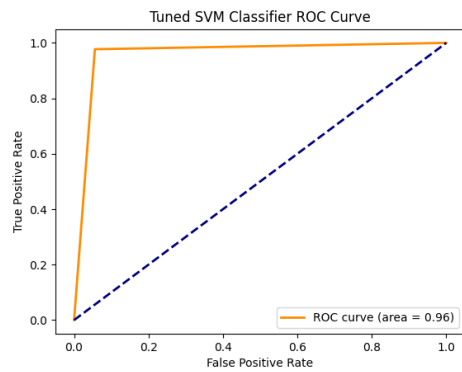


Figure 11: Existing SVM RoC Curve

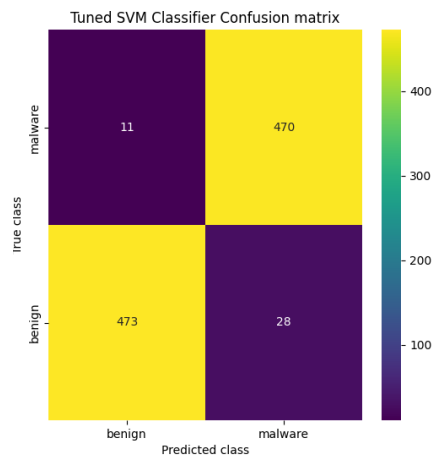


Figure 12: Existing SVM Confusion Matrix.

Figure 13 shows the Proposed VGG19 with ETC ROC Curve, which evaluates the Extra Trees Classifier (ETC) model combined with VGG19 features. The orange ROC curve achieves an AUC of 0.99, demonstrating exceptional classification performance, nearly perfect discrimination between benign and malware classes. The dashed blue line (AUC = 0.5) underscores the model's superior performance, with the curve closely aligned with the top-left corner, indicating high sensitivity and specificity.

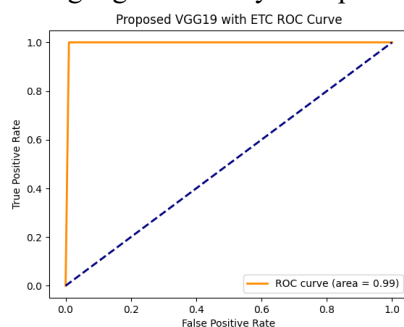


Figure 13: Proposed ETC RoC Curve.

Figure 14 shows the Proposed VGG19 with ETC Confusion Matrix, which details the ETC model's results. The 2x2 matrix shows 484 true negatives (benign correctly predicted as benign), 5 false positives (benign predicted as malware), 0 false negatives (no malware predicted as benign), and 493 true positives (malware correctly predicted as malware). This indicates near-perfect classification, with no missed malware instances and minimal misclassification of benign samples. Figure 15 shows the Prediction From Test Images, displaying the classification results of two test images. The images, which appear as grayscale patterns typical of byte-to-image mappings, are labeled with text overlays. The left image is classified as "benign" with the text "Output Classified as: benign" in green, while the right image is classified as "malware" with the text "Output Classified as: malware" in green. This visualization confirms the model's ability to predict classes for new inputs based on extracted features.

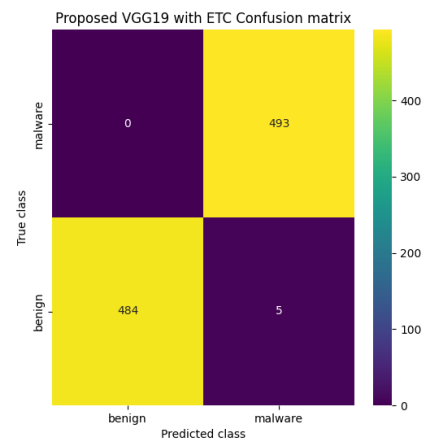


Figure 14: Proposed ETC Confusion Matrix.

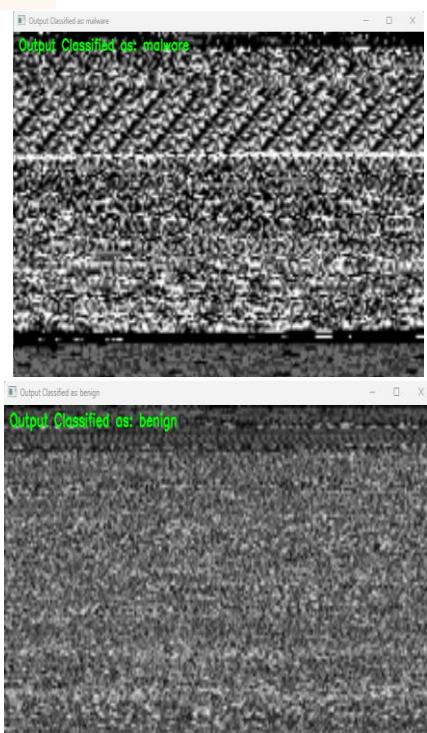


Figure 15: Prediction From Test Images.

CONCLUSION

The comparative evaluation of the proposed VGG19 with Extra Trees Classifier (ETC) against existing approaches such as LDA, QDA, and Tuned SVM demonstrates a significant improvement in malware detection accuracy when malware is represented as images. The proposed model achieves an impressive accuracy of 99.49%, outperforming LDA (97.04%), QDA (95.92%), and Tuned SVM (96.02%). Precision and recall values for VGG19 with ETC stand at 99.49% and 99.48%, respectively, showcasing its ability to minimize both false positives and false negatives effectively. Moreover, the AUC score of the proposed method (99.48%) indicates superior discriminative capability between benign and malware classes compared to LDA (97.07%), QDA (96.11%), and Tuned SVM (96.06%). These metrics reflect the robustness of deep feature extraction via VGG19 and the strong classification capability of ETC. In terms of class-specific performance, the proposed method shows a specificity of 100% and a sensitivity of 98.97%, ensuring that

benign applications are perfectly classified without false alarms, while maintaining extremely high detection for malware samples. Compared to LDA's sensitivity (95.59%) and specificity (98.55%), QDA's sensitivity (98.68%) and specificity (93.53%), and Tuned SVM's sensitivity (94.41%) and specificity (97.71%), the proposed approach delivers consistent superiority. The results affirm that the proposed methodology provides both high detection accuracy and reliability, making it an ideal solution for real-world cybersecurity applications.

References

- [1]. Song, Yafei, Dandan Zhang, Jian Wang, Yanan Wang, Yang Wang, and Peng Ding. "Application of deep learning in malware detection: a review." *Journal of Big Data* 12, no. 1 (2025): 99.
- [2]. Ferdous, Jannatul, Rafiqul Islam, Arash Mahboubi, and Md Zahidul Islam. "A Survey on ML Techniques for Multi-Platform Malware Detection: Securing PC, Mobile Devices, IoT, and Cloud Environments." *Sensors (Basel, Switzerland)* 25, no. 4 (2025): 1153.
- [3]. Du, Yao, CaiXia Gao, Xi Chen, MengTian Cui, LiLi Xu, and AoJi Ning. "Mobile malware detection method using improved GhostNetV2 with image enhancement technique: Y. Du et al." *Scientific Reports* 15, no. 1 (2025): 25019.
- [4]. Hawana, Ahmed, Emad S. Hassan, Walid El-Shafai, and Sami A. El-Dolil. "Enhancing malware detection with deep learning convolutional neural networks: Investigating the impact of image size variations." *Security and Privacy* 8, no. 2 (2025): e70000.
- [5]. Zhao, Yue, Farhan Ullah, Chien-Ming Chen, Mohammed Amoon, and Saru Kumari. "Efficient

- malware detection using hybrid approach of transfer learning and generative adversarial examples with image representation." *Expert Systems* 42, no. 2 (2025): e13693.
- [6]. Tariq, Muhammad Arham, Muhammad Ismaeel Khan, Aftab Arif, Muhammad Aksam Iftikhar, and Ali Raza A. Khan. "Malware Images Visualization and Classification with Parameter Tunned Deep Learning Model." *Metallurgical and Materials Engineering* 31, no. 2 (2025): 68-73.
- [7]. Zhao, Wenxiang, Juntao Wu, and Zhaoyi Meng. "Apppoet: Large language model based android malware detection via multi-view prompt engineering." *Expert Systems with Applications* 262 (2025): 125546.
- [8]. Alam, Inzamamul, Md Samiullah, S. M. Asaduzzaman, Upama Kabir, A. M. Aahad, and Simon S. Woo. "MIRACLE: Malware image recognition and classification by layered extraction." *Data Mining and Knowledge Discovery* 39, no. 1 (2025): 10.
- [9]. Zhang, Dandan, Yafei Song, Qian Xiang, and Yang Wang. "IMCMK-CNN: a lightweight convolutional neural network with multi-scale kernels for image-based malware classification." *Alexandria Engineering Journal* 111 (2025): 203-220.
- [10]. Nethala, Sainag, Pronoy Chopra, Khaja Kamaluddin, Shahid Alam, Soltan Alharbi, and Mohammad Alsaffar. "A Deep Learning-Based ensemble framework for robust Android malware detection." *IEEE Access* 13 (2025): 46673-46696.