Analyzing Texture-Based Feature Extraction and Classification Techniques for Malware Detection

PUNNA SHIVARANI
Post Graduate Student, M.C.A Department of Information Technology, Jawaharlal Nehru Technological University, Hyderabad,
punnashivarani@gmail.com

ABSTRACT

Malware detection is a vital part of cybersecurity, yet challenges such as low accuracy, high computation, and heavy resource usage persist. This work introduces a detection framework that integrates multiple texture-based feature extraction methods—SFTA, LBP, Haralick, Gabor, and Tamura—for analyzing malware images. Experiments were conducted on the MalImg and MaleVis datasets using various classifiers, including RF, KNN, GDA, SVM, LR, ELM,andan ensemble Voting Classifier combining Boosted Trees, Bagging with RF, and LR. The results show that GDA achieved 91.3% accuracy on MaleVis, while ELM reached 96.7% on MalImg. These findings emphasize the need for effective feature selection and classifier choice to improve malware detection performance.

KEYWORDS
Gabor, Gabor-KNN, GDA, LBP, Malimg, MaleVis dataset, malware detection, SFTA, SFTA-KNN,Tamura.

INTRODUCTION

The internet plays a central role in daily life, supporting activities such as communication, banking, shopping, and entertainment. With this increasing reliance on digital services comes a growing risk of cyberattacks. One of the most serious issues is the spread of malicious software, commonly known as malware, which threatens system integrity and user security. Malware includes a wide range of harmful programs such as viruses, worms, trojans, ransomware, bots, rootkits, potentially unwanted programs (PUPs), and other intrusive applications. Addressing these threats requires effective malware detection, a process that typically involves two stages: identifying whether software is malicious and classifying it into its specific family.

LITERATURE REVIEW

1. Traditional ML models (KNN, RF, DT, LR, etc.) achieve high accuracy, with Random Forest yielding **97.68%** on the UNSWNB15 dataset.
2. CNN-based frameworks (e.g., VBDN) improve generalization and classification, achieving **over 90% accuracy** across multiple datasets.
3. InceptionV3-based transfer learning with textural features reported **98.57% (MalImg)** and **97.78% (Microsoft BIG)** accuracy.
4. Red-channel analysis of color images outperformed grayscale, with Extra Trees reaching **98.37% accuracy**.
5. Autoencoder-enhanced CNNs with SDN honeypots achieved **98.50% accuracy** and **fast detection (0.006s)** for IIoT environments.

METHODOLOGY

Dataset Description:

Dataset : MalImg ,MaleVis

Preprocessing steps:

- Data Acquisition
- Image Processing
- Training and Testing
- Model Development
- Prediction

Model Architecture

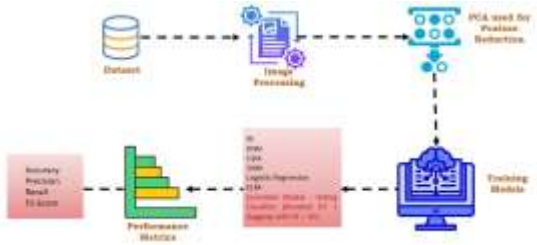


Fig. 1: System Architecture

Converts malware binaries into **grayscale images** and applies **texture descriptors** (SFTA, LBP, Haralick, Gabor, Tamura) to capture micro and macro-level patterns.Features are classified using **RF, KNN, GDA, SVM, LR, and ELM**, with an **ensemble model (Boosted Trees + Bagging with RF + LR)** to boost accuracy and robustness.Evaluation metrics include **Accuracy, Precision, Recall, and F1-Score** (visualized in bar charts).

Dataset Collection:

MaleVis → Binary classification (malware vs. non-malware).

MalImg → Multi-class classification (backdoor, trojan, worm, etc.).

Pre-processing Steps:

- **Image Processing:** Resize, grayscale conversion, NumPy array transformation, label encoding.
- **Feature Extraction:** Texture-based descriptors (SFTA, LBP, Haralick, Gabor, Tamura).
- **Scaling:** Normalization/standardization for equal feature contribution.
- **PCA:** Reduces dimensionality, improves efficiency, and avoids overfitting.

System Setup

- **Development Environment:** Anaconda
- **Programming Language:** Python
- **Web Framework (Frontend):** Flask
- **Backend Environment:** Jupyter Notebook
- **Database Management System:** SQLite3
- **Frontend Technologies:** HTML, CSS, JavaScript, and Bootstrap 4
- **Operating System:** Windows (any compatible version)
- **Processor:** Intel Core i5 or higher
- **Memory (RAM):** Minimum 8 GB
- **Storage:** At least 25 GB of free space on the local drive

RESULTS

Table.1 Performance Evaluation – MalImg Dataset

ML Model	Accuracy	Precision	Recall	F1 Score
Random Forest	0.957	0.963	0.957	0.959
KNN	0.501	0.803	0.501	0.417
GDA	0.933	0.948	0.933	0.937
SVM	0.966	0.967	0.966	0.966
Logistic Regression	0.965	0.966	0.965	0.965

ELM	**0.967**	**0.970**	**0.967**	**0.968**
Ensemble	0.965	0.967	0.965	0.966

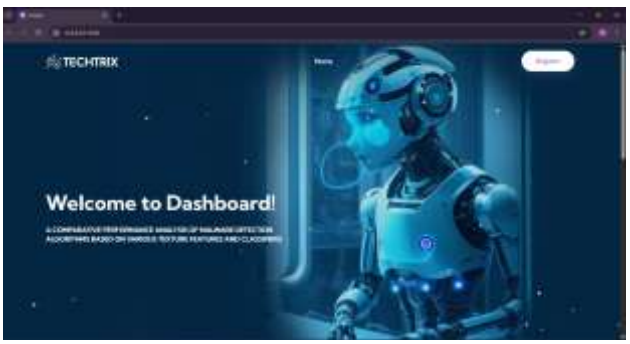
Table.2 Performance Evaluation – MaleVis Dataset

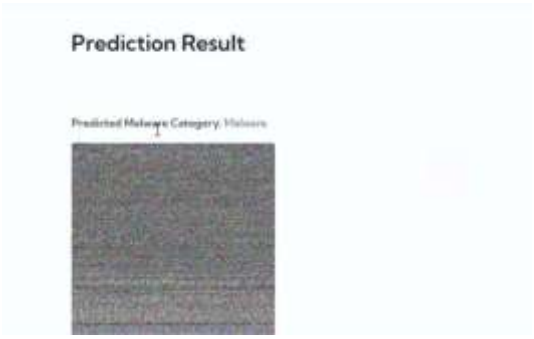
ML Model	Accuracy	Precision	Recall	F1 Score
Random Forest	0.897	0.979	0.897	0.929
KNN	0.839	0.841	0.839	0.819
GDA	**0.913**	**0.955**	**0.913**	**0.928**
SVM	0.904	0.909	0.904	0.906
Logistic Regression	0.912	0.935	0.912	0.921
ELM	0.880	0.871	0.880	0.871
Ensemble	0.910	0.960	0.910	0.928

Graph.1 Comparison Graph – MalImg Dataset

Graph.2 Comparison Graph - MaleVis Dataset

OUTPUT SCREENS





CONCLUSION

The proposed framework effectively combines texture feature fusion (SFTA, LBP, Haralick, Gabor, Tamura) with ensemble learning for accurate malware detection. GDA achieved 91.3% on MaleVis, while ELM reached 96.7% on MalImg, highlighting their strengths on different datasets. The system ensures high accuracy, scalability, and suitability for real-time cybersecurity. Future work will explore deep learning (CNNs), integration of dynamic behavioral features, large-scale testing, and explainable AI for improved interpretability.

REFERENCES

[1] Azeem, M., Khan, D., Iftikhar, S., Bawazeer, S., & Alzahrani, M. (2024). Analyzing and comparing the effectiveness of malware detection: A study of machine learning approaches. *Heliyon*, *10*(1).

[2] Zhong, F., Hu, Q., Jiang, Y., Huang, J., Zhang, C., & Wu, D. (2024). Enhancing Malware Classification via Self-Similarity Techniques. *IEEE Transactions on Information Forensics and Security*.

[3] Khan, F. B., Durad, M. H., Khan, A., Khan, F. A., Rizwan, M., & Ali, A. (2024). Design and performance analysis of an anti-malware system based on generative adversarial network framework. *IEEE Access*, *12*, 27683-27708.

[4] Liu, Y., Fan, H., Zhao, J., Zhang, J., & Yin, X. (2024). Efficient and generalized image-based CNN algorithm for multi-class malware detection. *IEEE Access*.

[5] Kumar, S., Sapru, Y., & Chahal, N. S. (2024, August). Malicious program detection using distinct textural features and fine-tuned InceptionV3 model. In *2024 IEEE 5th India Council International Subsections Conference (INDISCON)* (pp. 1-6). IEEE.