World Journal of Advanced Research and Reviews
eISSN: 2581-9615 CODEN (USA): WJARAI
Cross Ref DOI: 10.30574/wjarr
Journal homepage: https://wjarr.com/

(RESEARCH ARTICLE)

Check for updates

DEVELOPMENT OF A HYBRID NN–CNN DEEP LEARNING FRAMEWORK FOR INTELLIGENT MALWARE DETECTION, FAMILY CLASSIFICATION, AND VARIANT IDENTIFICATION

Chioma Grace Nwankwo [1,*], Bethran Chibuike Amanze [2] and Ikechukwu Amaefule [2]

[1] Computer Science department, College of Physical and Applied Sciences, Michael Okpara University of Agriculture, Umudike.
[2] Computer Science department, Faculty of Physical Sciences, Imo State University Owerri, Imo State.
* Corresponding Author: Chioma Grace Nwankwo, chiomanwankwo740@gmail.com

World Journal of Advanced Research and Reviews, 2026, 31(02), 954–964

Publication history: Received on 30 June 2026; revised on 16 August 2026; accepted on 18 August 2026

Article DOI: https://doi.org/10.30574/wjarr.2026.31.2.2132

ABSTRACT

Malware continues to be one of the major cybersecurity threats affecting individuals, businesses, government institutions, and critical infrastructure. The problem has become more difficult because modern malware can change its structure, hide important characteristics, and produce multiple variants while maintaining similar malicious behaviour. Traditional signature-based detection methods remain useful for known threats, but they may struggle when presented with new or modified malware. This has encouraged researchers to explore artificial intelligence (AI) and deep learning as alternative approaches to automated malware analysis. This study proposes a Hybrid Neural Network–Convolutional Neural Network (NN–CNN) Deep Learning Framework for malware detection, malware-family classification, and malware-variant identification. The proposed framework combines two complementary learning approaches. The NN component will learn patterns from structured malware features, while the CNN component will extract spatial patterns from visual representations of malware binaries. The learned representations will then be combined through a feature-fusion mechanism and used for hierarchical classification. In the proposed approach, the system will first determine whether a file is benign or malicious, then identify the malware family, and finally attempt to determine the specific variant or subfamily. The proposed framework also considers two important issues in practical malware detection: model explainability and generalization to previously unseen malware. Explainable AI techniques will be investigated to help security analysts understand the factors influencing model decisions, while cross-dataset and temporal evaluations will be used to assess whether the model can maintain its performance beyond the dataset on which it was trained. The study is expected to contribute a more comprehensive approach to intelligent malware analysis by bringing detection, family classification, variant identification, explainability, and robustness into a single research framework.

Keywords: Malware Detection, Malware Variants, Neural Networks, Convolutional Neural Networks, Deep Learning, Cybersecurity.

1 INTRODUCTION

Malware has evolved considerably over the past few decades. What was once largely associated with relatively simple computer viruses has developed into a broad ecosystem involving ransomware, trojans, spyware, worms, botnets, information stealers, rootkits, backdoors, and other malicious programs. The increasing number and diversity of malware samples make it increasingly difficult for cybersecurity professionals to examine every suspicious file manually.

One of the traditional approaches to malware detection is signature-based detection. In this approach, a known malware sample is analysed and a unique signature is generated. Future files can then be compared against the signature to determine whether they are malicious. Although this approach remains valuable, it has an important weakness: it is generally most effective when the threat has already been identified. Malware developers can modify existing programs through techniques such as obfuscation, packing, encryption, and mutation, causing different versions of essentially related malware to appear substantially different. Gibert et al. (2019), for example, observed that malware authors frequently modify malicious files in order to evade detection, even when the modified samples retain similar malicious behaviour.

This problem has encouraged researchers to investigate machine learning and deep learning techniques for malware analysis. Rather than relying entirely on predefined signatures, these approaches can learn patterns from previously observed data. Recent reviews indicate that deep-learning approaches have been applied to malware detection across several operating-system environments and using different types of representations, including text, behavioural data, binary information, and malware images.

Among these approaches, Convolutional Neural Networks (CNNs) have attracted considerable attention. One important idea is to transform an executable binary into an image and allow a CNN to learn visual patterns associated with different malware families. Gibert et al. (2019) demonstrated the feasibility of using CNNs to classify malware represented as images, showing how visual representations can capture similarities between malware samples.

However, malware detection is more complicated than simply deciding whether a file is malicious. In a practical cybersecurity environment, an analyst may also want to know which family the malware belongs to and whether it represents a particular variant of that family. This distinction is important because identifying a malware family can provide useful information about its likely behaviour, while variant identification can provide a more detailed understanding of the threat.

Recent research has moved toward hierarchical malware analysis in which a system performs several related tasks, including benign/malicious detection, family classification, and subfamily identification. A recent CNN-based hierarchical study, for example, investigated three levels of classification using executable-file representations: malware detection, family classification, and subfamily assignment.

Despite these developments, important challenges remain. Deep-learning models may perform well on benchmark datasets but experience reduced performance when exposed to unseen data or changing malware distributions. Adversarial manipulation is another concern because attackers may intentionally modify samples to mislead machine-learning systems. A recent survey by Bensaoud et al. (2024) identifies generalization, adversarial attacks, dataset limitations, and explainability among the continuing challenges in deep-learning-based malware detection.

These challenges provide the motivation for the present study. Rather than developing another standalone CNN classifier, this research proposes a **hybrid NN–CNN architecture** that combines structured and visual representations of malware and uses them to perform multiple levels of malware analysis.

2 BACKGROUND OF THE STUDY

2.1 Malware Detection

Malware detection involves identifying whether a program or file contains malicious characteristics. Malware analysis is commonly divided into **static analysis** and **dynamic analysis**.

Static analysis examines a program without executing it. Information such as executable headers, imported libraries, API calls, strings, sections, byte sequences, and entropy can be extracted and used as features.

Dynamic analysis, on the other hand, examines what happens when a suspicious program is executed in a controlled environment. Researchers may observe API calls, file-system activity, network connections, registry modifications, process behaviour, and other runtime activities.

A recent review of deep-learning malware detection techniques notes that static and dynamic analysis remain two major approaches for extracting information from malware samples.

Both approaches have strengths and weaknesses. Static analysis can be faster because the sample does not need to be executed, but sophisticated malware can hide important information through packing and obfuscation. Dynamic analysis can reveal behavioural information that may not be visible during static examination, but it can require additional computational resources and a secure execution environment.

The proposed study will therefore investigate how different forms of malware information can be represented and combined within a deep-learning framework.

2.2 Neural Networks in Malware Detection

Artificial Neural Networks (NNs) are computational models inspired by the way biological neurons process information. In machine learning, an NN learns relationships between input features and desired outputs through a process of adjusting its internal parameters during training.

For malware analysis, a neural network can receive structured information extracted from an executable file and learn combinations of features that distinguish benign files from malicious ones.

For example, a model may learn that a particular combination of:

- high section entropy;
- unusual imported APIs;
- suspicious executable sections;
- abnormal file characteristics; and
- specific behavioural indicators

is associated with malicious software.

One advantage of neural networks is their ability to learn nonlinear relationships among features. This makes them potentially useful when the relationship between individual malware characteristics and malicious behaviour is not straightforward.

However, a conventional NN operating only on structured features may fail to capture some spatial patterns present in the raw binary representation of malware. This provides a reason for investigating a second learning branch based on CNNs.

2.3 Convolutional Neural Networks for Malware Classification

CNNs were originally developed primarily for image-processing tasks, but their ability to identify local patterns and construct increasingly complex representations has made them useful in other domains.

In malware research, an executable binary can be transformed into a visual representation. The resulting image may contain patterns that reflect structural characteristics of the original file.

Gibert et al. (2019) investigated CNN-based malware classification using image representations and demonstrated that malware belonging to related families can exhibit visual similarities.

Another line of research has used entropy-based representations to describe structural characteristics of malicious programs. Gibert et al. showed that entropy patterns could provide information useful for malware categorization using CNN-based approaches.

The important point is that a CNN does not need a researcher to manually specify every relevant visual pattern. Instead, the convolutional layers learn useful representations during training.

Nevertheless, CNN-based approaches also have limitations. A visual representation does not necessarily preserve every semantic characteristic of an executable file. Two files that look visually similar may have different behaviour, while

malware samples that belong to the same family can be modified enough to appear visually different. This is one reason why relying on a single representation may not be sufficient.

2.4 Rationale for a Hybrid NN–CNN Framework

The central idea behind the proposed research is that structured and visual malware information can complement each other.

The NN branch will focus on structured characteristics, while the CNN branch will focus on visual or spatial characteristics.

Conceptually:

Structured Features → NN → Learned Representation

Binary/Image Features → CNN → Learned Representation

↓

Feature Fusion

↓

Classification

The NN branch may capture relationships among PE-header characteristics, API information, entropy, strings, opcode statistics, and other structured features.

The CNN branch may capture spatial patterns associated with binary or image representations.

The two representations can then be combined to create a richer representation of the malware sample.

Recent research supports the broader idea of hybrid deep-learning approaches. For example, a 2024 study proposed a multi-level deep-learning approach that sequentially combined different deep-learning architectures for malware-family classification.

The proposed study extends this direction by specifically investigating the complementary use of NN and CNN representations and applying the resulting representation to multiple malware-analysis tasks.

2.5 Problem Statement

The increasing diversity and sophistication of malware present significant challenges to conventional cybersecurity systems. Malware authors can modify malicious programs in ways that make individual samples appear different while maintaining similar functionality. Consequently, detecting malware based solely on known signatures or isolated characteristics may not be sufficient for identifying new or modified variants.

Deep-learning techniques have provided promising results in malware detection and classification. However, existing approaches often concentrate on one specific task, such as binary malware detection or malware-family classification. In addition, the performance of deep-learning systems can be affected by dataset imbalance, distribution changes, unseen malware, adversarial manipulation, and limited interpretability.

Another important issue is the lack of transparency associated with many deep-learning systems. A model may correctly identify a sample as malicious without explaining why it reached that conclusion. In a cybersecurity environment, this can make it difficult for analysts to trust or validate automated decisions. Recent research on explainable AI for malware analysis identifies this lack of transparency as an important barrier to the practical adoption of AI-based malware detection.

Therefore, there is a need for a framework that does more than classify malware as simply "malicious" or "benign." The proposed research seeks to develop a hybrid system capable of detecting malware, identifying its family, investigating its variant, and providing interpretable information about the classification decision.

2.6 Aim of the Study

The aim of this research is to develop a hybrid NN–CNN deep-learning framework for intelligent malware detection, malware-family classification, and malware-variant identification, with particular attention to generalization, robustness, and explainability.

2.7 Research Objectives

The specific objectives are to:

- Examine existing AI and deep-learning approaches used for malware detection and classification.
- Identify suitable structured and visual representations of malware samples.
- Develop an NN model for learning structured malware characteristics.
- Develop a CNN model for learning patterns from malware visual representations.
- Develop a feature-fusion mechanism for combining the representations learned by the NN and CNN branches.
- Develop a hierarchical malware-analysis model for malware detection, family classification, and variant identification.
- Evaluate the ability of the proposed framework to generalize to previously unseen malware.
- Investigate the robustness of the proposed framework against selected malware modifications and adversarial conditions.
- Incorporate explainability techniques to provide meaningful interpretations of model predictions.
- Compare the proposed framework with standalone NN, standalone CNN, and conventional machine-learning approaches.

3 PROPOSED METHODOLOGY

The study will adopt a quantitative experimental research methodology.

The overall process will consist of:

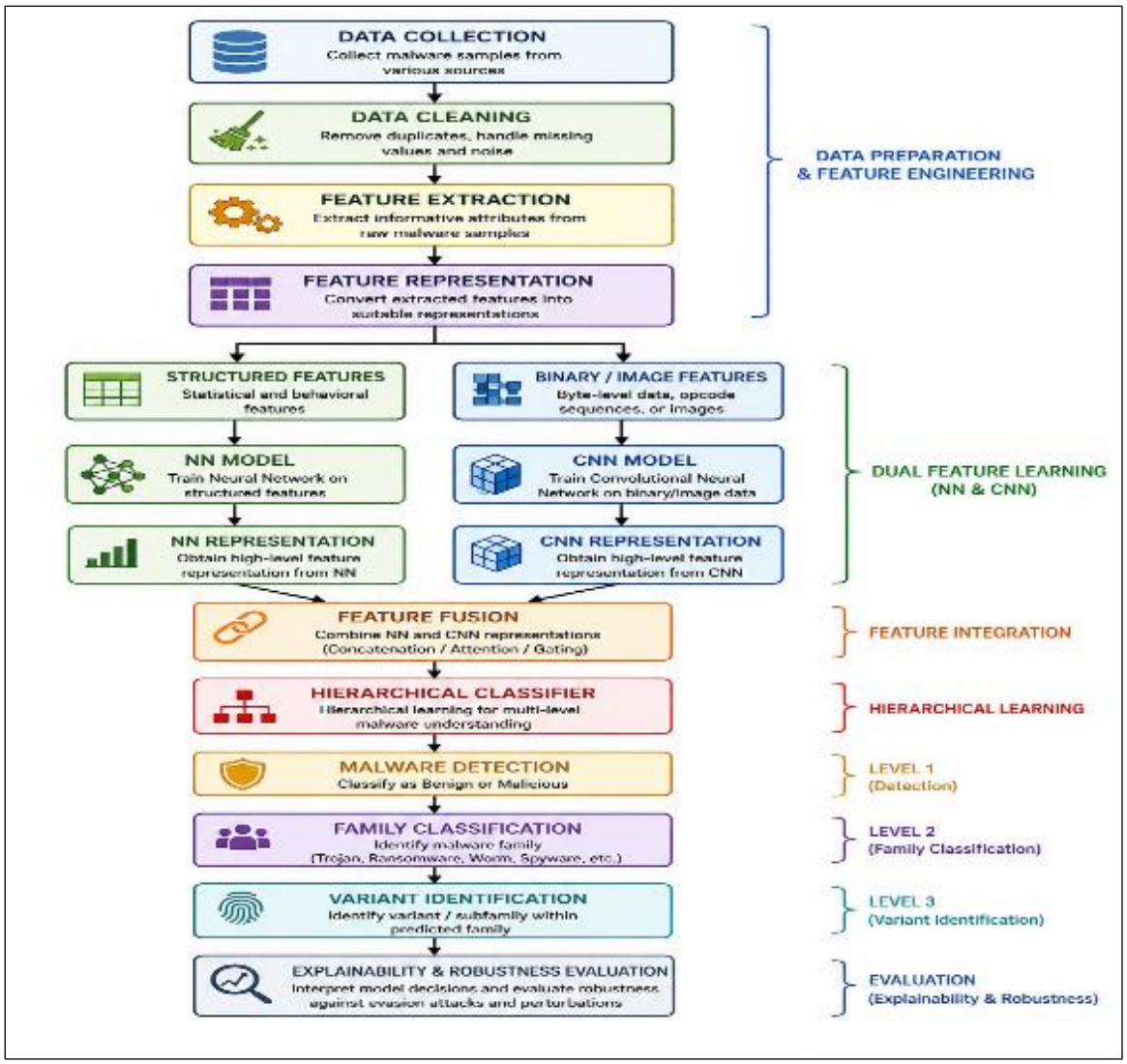

Figure 1: Proposed Architecture

3.1 Data Collection

The research will require both malware and benign software samples.

Potential datasets include established research datasets such as:

- Microsoft Malware Classification Challenge dataset;
- Malimg;
- MaleVis;
- EMBER; and
- other publicly available and appropriately licensed datasets.

The choice of dataset will depend on the research objectives and the availability of reliable labels.

Particular attention should be paid to the quality of the labels. Malware datasets may contain duplicate samples, mislabeled samples, or samples that are extremely similar to one another. If such issues are not addressed, the resulting model may appear more accurate than it actually is.

This is particularly important because standard benchmark datasets for malware research remain difficult to establish, according to recent reviews.

3.2 Data Preprocessing

The preprocessing stage will involve:

- Removing corrupted samples.
- Identifying and handling duplicate files.
- Verifying malware-family labels where possible.
- Handling missing feature values.
- Normalizing numerical features.
- Encoding categorical features.
- Addressing class imbalance.
- Converting suitable executable files into image representations.

For image-based analysis, the binary content can be converted into grayscale representations. The resulting images can then be resized or normalized before being provided to the CNN.

However, care must be taken to ensure that preprocessing does not introduce information leakage between training and testing datasets.

3.3 Proposed NN Architecture

The NN branch will process structured malware features.

A conceptual architecture may be:

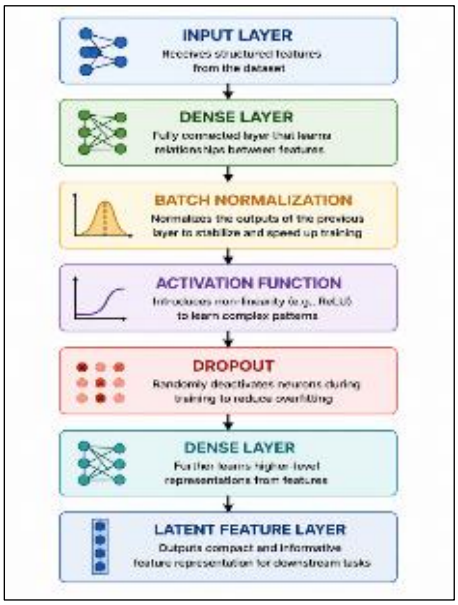

Figure 2: Neural Network (NN)

The Neural Network (NN) branch of the proposed framework is responsible for learning from the structured features extracted from malware and benign software.

The process starts with the Input Layer, which receives information such as PE-header details, entropy, API calls, file characteristics, and other extracted features. The data then passes through a Dense Layer, where the network begins to identify relationships between these features.

Batch Normalization helps stabilize the learning process, while the Activation Function, such as ReLU, allows the network to learn more complex and non-linear patterns. Dropout is then used to reduce overfitting and help the model perform better when it encounters new malware samples.

A second Dense Layer further processes the learned information and identifies higher-level patterns. Finally, the Latent Feature Layer produces a compact representation of the malware characteristics learned by the NN.

In the proposed hybrid system, this NN representation is combined with the CNN feature representation through feature fusion. The combined information is then used for malware detection, family classification, and variant identification.

The final latent layer will provide the feature representation used in the fusion stage.

The exact number of layers and neurons will be determined experimentally rather than assumed in advance.

3.4 Proposed CNN Architecture

The CNN branch will process malware image representations. A conceptual structure is:

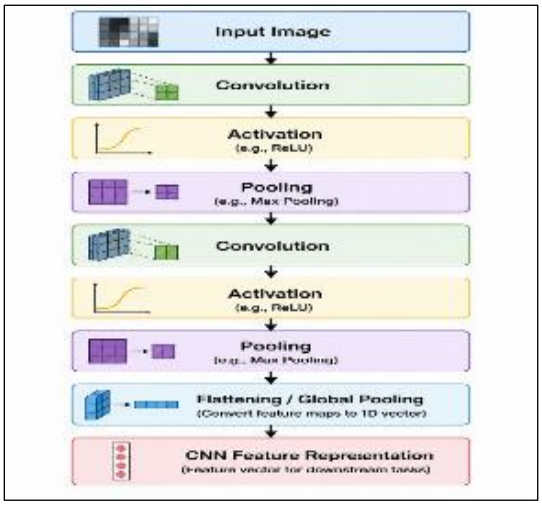

Figure 3: CNN architecture

he CNN component of the proposed Hybrid NN–CNN malware detection framework is designed to extract meaningful patterns from **malware binary images**.

The process begins by converting a malware binary into an **image representation**, which is then provided to the CNN as the input. The image passes through successive **convolution, activation, and pooling layers**. Convolution extracts important patterns, activation functions such as ReLU introduce non-linearity, and pooling reduces the size of the feature maps while retaining important information.

After these operations, **flattening or global pooling** converts the learned feature maps into a one-dimensional feature vector. This becomes the **CNN Feature Representation**, which captures important characteristics learned from the malware image.

The CNN representation is then combined with the **NN representation**, which is obtained from structured malware features. Through **feature fusion**, both representations are integrated and passed to the hierarchical classifier

The CNN architecture may initially be implemented using a relatively lightweight architecture before being compared with deeper alternatives.

Transfer learning may also be investigated where appropriate.

3.5 Model Evaluation

The proposed framework will be evaluated using several performance measures: **Accuracy, Precision, Recall.** The three measures provide different views of the model's performance:

Measure	Main Question	Importance
Accuracy	How many predictions were correct overall?	Shows overall performance
Precision	How many predicted malware samples were actually malware?	Measures false-alarm control
Recall	How much of the actual malware did we detect?	Measures ability to catch threats

3.6 Expected Contribution of the Research

The proposed research is expected to contribute in several areas.

- First, it will provide a hybrid learning architecture that combines structured and visual malware representations.
- Second, it will investigate a multi-level malware-analysis process rather than limiting the task to benign/malicious classification.
- Third, it will investigate whether feature fusion can improve the identification of malware families and variants.
- Fourth, it will provide a more rigorous evaluation of model generalization by including unseen and temporally separated malware.
- Fifth, it will examine the robustness of the model under malware modification.

Finally, it will investigate the use of explainability techniques to make the resulting predictions more understandable to cybersecurity analysts.

4 DISCUSSION

The main argument of this research is that malware contains different types of information, and no single representation necessarily captures all of it.

A structured feature representation can provide information about the internal characteristics of a file. A visual representation, on the other hand, can reveal spatial patterns that may be useful for distinguishing malware families. CNN-based research has already demonstrated that malware images can contain meaningful family-level information.

The proposed framework brings these two perspectives together.

However, the success of the framework should not be judged solely by its accuracy on a single benchmark. A model that obtains very high accuracy on familiar samples but performs poorly on new malware would have limited practical value.

This is why generalization, robustness and explainability are included as core parts of the research rather than being treated as optional additions. Recent literature consistently identifies these issues as important challenges for AI-based malware detection.

Explainability is particularly important. Security analysts need to make decisions based on automated alerts, and a system that simply says "malicious" without providing supporting evidence may be difficult to trust. Research on explainable malware detection has therefore emphasized the need to make AI predictions more transparent and interpretable.

5 CONCLUSION

Malware continues to evolve, making automated and intelligent detection increasingly important. Although traditional signature-based approaches remain useful, they may struggle with new, modified, and obfuscated malware. Deep learning provides an opportunity to learn complex malware patterns directly from data, and CNN-based approaches have demonstrated that visual representations of executable files can be useful for malware classification.

This research proposes a Hybrid NN–CNN Deep Learning Framework for Intelligent Malware Detection, Family Classification, and Variant Identification. The framework combines structured malware characteristics with visual representations and uses feature fusion to create a more comprehensive representation of each sample.

The proposed system is designed around three major levels of analysis: detecting whether a file is malicious, identifying the malware family, and determining its variant or subfamily. Beyond these classification tasks, the research will investigate generalization to unseen malware, robustness against malware modification, and explainability.

The significance of the proposed framework lies not simply in achieving high classification accuracy but in developing a malware-analysis system that is more comprehensive, explainable, and relevant to changing cybersecurity environments. If successfully validated, the framework could contribute to the development of intelligent malware-analysis systems capable of supporting security analysts in identifying and understanding increasingly complex malware threats.

Research Gap

Based on the reviewed literature, four major gaps can be identified.

- **Gap 1: Single-representation learning**

Many existing systems depend primarily on one type of representation, such as binary images or structured static features. While each representation provides useful information, relying on one representation may limit the information available to the classifier.

- **Gap 2: Limited multi-level classification**

A significant amount of research focuses on malware detection or family classification independently. However, a practical malware-analysis system should ideally support several levels of analysis.

Recent work demonstrates the feasibility of hierarchical classification involving malware detection, family classification and subfamily assignment, indicating that this direction is becoming increasingly relevant.

- **Gap 3: Generalization to unseen malware**

High accuracy on a benchmark dataset does not necessarily mean that a model will perform well against new malware. Deep-learning surveys have highlighted the difficulty of evaluating generalization and the effect of adversarial attacks on unseen samples.

- **Gap 4: Explainability**

Many malware classifiers operate as black boxes. Security analysts may therefore receive a prediction without sufficient information about the reasons behind it. Recent research specifically recommends greater use of XAI techniques in malware analysis.

These gaps motivate the development of the proposed hybrid framework.

STATEMENTS ON COMPLIANCE WITH ETHICAL STANDARDS

Acknowledgments

We acknowledge the services of Charles Osedeke and Beloved Obinnaya for ensuring that all necessary grammatical corrections were made. We also appreciate the MOUAU librarian who granted us access to materials used in the work.

Disclosure of conflict of interest

No conflict of interest to be disclosed.

REFERENCES

[1] Bensaoud, A., Kalita, J., & Bensaoud, M. (2024). A survey of malware detection using deep learning. Machine Learning with Applications, 16, 100546.

[2] Card, Q., Simpson, D., Aryal, K., Gupta, M., & Islam, S. R. (2024). Explainable deep learning models for dynamic and online malware classification. arXiv.

[3] Gibert, D., Mateu, C., Planes, J., & Vicens, R. (2019). Using convolutional neural networks for classification of malware represented as images. Journal of Computer Virology and Hacking Techniques, 15, 15–28.

[4] Gibert, D., Mateu, C., Planes, J., & Vicens, R. (2018). Classification of malware by using structural entropy on convolutional neural networks. Proceedings of the AAAI Conference on Artificial Intelligence, 32(1).

[5] Manthena, H., Shajarian, S., Kimmell, J., Abdelsalam, M., Khorsandroo, S., & Gupta, M. (2024/2025). Explainable Artificial Intelligence (XAI) for Malware Analysis: A Survey of Techniques, Applications, and Open Challenges. arXiv.

[6] A review of deep learning based malware detection techniques. (2024). Neurocomputing, 598, 128010.

[7] Deep hybrid approach with sequential feature extraction and classification for robust malware detection. (2024). Egyptian Informatics Journal, 27, 100539.

[8] Hierarchical malware detection, family identification, and variant attribution using CNN-based hybrid models on grayscale executable images. (2026). Scientific Reports.

[9] Explainability in AI-based behavioral malware detection systems. (2024). Computers & Security, 141, 103842.