A Malware Detection and Classification using Neural Networks: A Review

Mohammed Abosaeeda [1] , Mahmud Mansour [2]

[1] Computer Technology Department, Higher Institute For Science and Technology AL-Garabolli
m.abosaeeda@uot.edu.ly
[2] Department of Computer Networks, University of Tripoli
mah.mansour@uot.edu.ly

Abstract. The rapid evolution of malware, particularly polymorphic and metamorphic variants, has rendered traditional detection methods, such as signature-based and behavioural detection, increasingly ineffective. This paper's objective is a comprehensive review of Artificial Neural Networks (ANNs) for malware detection and classification via a comprehensive review of the most widely used ANNs. The study focuses on supervised models, unsupervised models, and hybrid architectures across diverse environments. The study results indicate that the supervised models achieve exceptional accuracy (>95%); the unsupervised models offer interpretability and adaptability to evolving threats but face challenges in generalising to unseen data. Conversely, hybrid models combine spatial and temporal feature extraction, achieving 99.4% accuracy, albeit with higher computational costs. This study emphasises the importance of the need for robust frameworks against obfuscation, efficient architectures for resource-constrained environments, and enhanced generalisation across malware families.

Keywords: Malware Detection, Malware Classification, Malware image**,** Artificial neural networks algorithms.

1 Introduction

The ever-evolving cyber threats have made malware detection a critical area of research in cybersecurity. Traditional detection methods, including signature-based and heuristic methods, have been extensively used to combat malware. However, these methods face significant limitations when dealing with advanced malware variants, such as polymorphic and metamorphic malware, which can change their code structure to evade detection. Malware, a term that encompasses various forms of malicious software such as viruses, worms, ransomware, and Trojans, is designed to infiltrate, damage, or exploit systems. Malware that compromises data and systems due to malicious attacks and threats must be confronted.

Figure 1 shows the top three sources of attacks, top three targets of attacks, types of attacks, the attacker IP addresses, the attacker location statistics, the countries attacked by malware, and the major sources of malware in 2025 [1]. Malware is rapidly growing

all over the world, malware types vary in purpose and intent, nevertheless all types of malware cause damage.

According to studies, they have caused several different types of damage. Traditional detection methods, including signature-based and behavioral methods, have been widely used to combat malware. However, these methods face substantial limitations when dealing with advanced malware variants, such as polymorphic and metamorphic malware, which can change their code structure to evade detection methods.

The study in [2] indicated that malware could harm all types of sensitive devices and data by various means, including unauthorized access, which infringes on the rights of their owners due to the security vulnerabilities exploited by malware creators and cybercriminals. According to the "China Network Security Report 2021", Rising's "Cloud Security" system intercepted 119 million virus samples, with 259 million virus infections found, and the total number of viruses decreased relatively compared to 2020.

As cyber threats become increasingly complex and frequent, traditional malware analysis, detection and classification methods, such as static and dynamic analysis, signature-based detection and behavioural detection, have become more difficult, and their limitations against advanced, polymorphic and metamorphic malware, which constantly changes its code to evade detection techniques, require the search for innovative methods that leverage the power of modern technologies [3,4].

Fig. 1. Worldwide attacks.

Recent developments in the various branches of artificial intelligence, including machine learning and deep learning, more specifically in artificial neural networks (ANNs), have opened new horizons to address the limitations in traditional methods of detecting, and classifying malware. Neural networks have shown great performance in several areas, particularly in the field of cybersecurity, by detecting malware through learning patterns in datasets, whether it is a dataset, such as comma-separated values, (CSV) files or image data representing malware by converting them into binary, grayscale or colored images of different types of malwares. The potential of artificial neural networks (ANNs) is not only to enhance detection accuracy, but also to reduce false positive rates, which are a critical factor in maintaining effective and reliable cybersecurity systems.

Since it generalizes well to unseen data, this paper surveys traditional methods and their shortcomings, including malware detection methods (static and dynamic, signature-based detection, and behavioral detection), machine learning, and deep learning, focusing primarily on artificial neural networks (ANNs) in several environments.
We evaluate the strengths and weaknesses of neural network models for malware detection. Recent research uses artificial neural networks (ANNs) to detect and classify malware and achieves significant performance.

2 Artificial Intelligence (AI)

Artificial intelligence is defined as systems, applications, or computer models integrated into machines to perform tasks that simulate how human intelligence works. A number of technologies that enable machines to simulate real intelligence consist of different sub-sections (machine learning (ML), deep learning (DL), neural networks (NN) as shown in the Figure 2. Applied in several fields such as education, health, problem solving, and decision-making [5]. In the field of cybersecurity, it has made great progress in protecting systems, networks and servers. It has been included in cybersecurity systems to perform protection tasks and provide information to human security teams in identifying and discovering security threats and responding to them. Through its use, it has reduced risks with high efficiency in processing large amounts of security data [6].

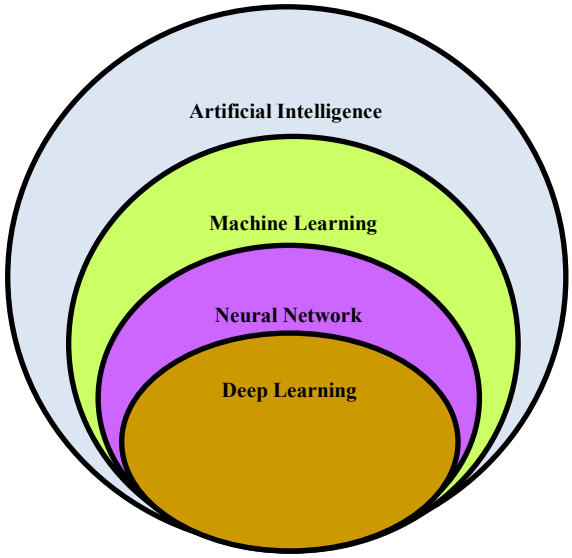

Fig. 2. Artificial Intelligence Paradigms

Artificial Intelligence is the ability of a computer program to function like a human mind on levels:

- Narrow AI: Can perform only a specific task or specific problems,

- General AI: When it can perform any intellectual task with human capacity across a wide range of tasks,
- Autonomous AI: Systems at this level are designed to make decisions independently in complex environments [6].

The AI techniques used in cybersecurity, particularly in malware detection and response, as illustrated in Figure 2. Including several approaches:

2.1 Machine Learning (ML)

It is one of the technologies and contains a set of algorithms that enabling machines to learn from a dataset. This allows for improved detection of malware and its classification according to specific mechanisms, such as its degree of danger or the family to which it belongs, etc.
In [7] the Machine learning techniques are divided into supervised, unsupervised and semi-supervised learning models, as illustrated in Figure 3. These are the types of machine learning algorithms.

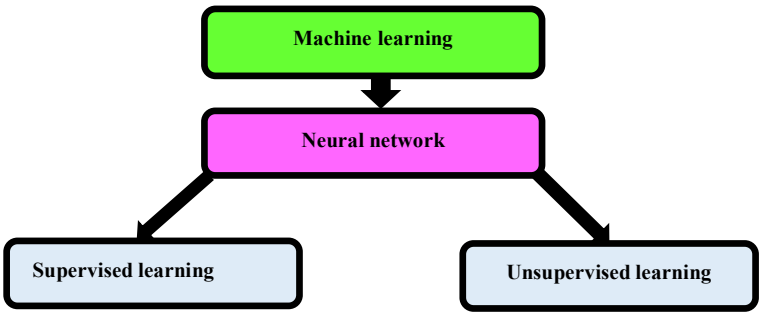

Fig. 3. Types algorithms of neural network

2.1.1 Supervised Machine Learning

Supervised learning is a target function derived from a labelled training dataset. The function is developed from input (x) to output (y) by analyzing the data. The output includes the labelling of the input data, which is the information needed for the model to make correct discoveries and predictions.
Supervised algorithms are applied in classification and regression tasks. Some of these algorithms are k-NN algorithms, decision trees, and support vector machines (SVM). Supervised learning relies on having a training dataset for each supervised data point, which is correctly labeled. The model is evaluated by validating the trained model and testing it on a different test dataset that was not used for training [7].

2.1.2 Unsupervised Machine Learning

Unsupervised learning involves learning from unlabeled data (without labels); where only input data (x) is available and there is no specific output data. This type of learning

focuses on the basic structure or distribution of data to recognise patterns or understand the relationship between them. Models of this type are organised without external supervision and are often used in clustering and understanding the rules of association between data. Among these algorithms are k-means, hierarchical clustering, and principal component analysis (PCA). This learning is also applied in detecting unusual behaviour in the field of cybersecurity, analysing large data sets to facilitate their understanding and interpretation, and extracting features from data [7].

2.1.3 Semi-Supervised Machine Learning

This type of machine learning uses a training dataset for models consisting of a mixture of labelled and unlabeled data by combining supervised and unsupervised learning. This is because unlabeled data provides valuable information about the data and its distribution, and the amount of this data is often larger than that labelled data. Among these techniques are graph-based models and generative models, which are used in cybersecurity to detect intrusions [7].

2.2 Neural Network (NN)

It is a computational model of artificial intelligence inspired by the work of the human brain, consisting of interconnected elements called nodes or neurons. It processes information, reduces errors, and learns patterns from experience in order to make decisions based on input data. It is used in various fields, such as image recognition [8].

2.3 Deep learning (DL)

A set of neural network algorithms that mimic the human brain to solve complex problems by recognizing patterns, capturing concepts from processing large amounts of data, and learning from them to gain knowledge [9].

3 Artificial Neural Networks (ANNs)

Neural networks are a crucial aspect of AI and ML and reside somewhere between deep learning and machine learning. These computational systems are based on the biological neural network present in the human brain. Networks of nodes known as neuronees are placed into layers that are interconnected. Among such networks are convolutional neural networks (CNNs) and recurrent neural networks (RNNs). Widespread applications of neural networks in the analytics of large datasets for the purposes of recognition of patterns and learning from them, as well as in cybersecurity for event detection, invisible vulnerabilities and, polymorphic viruses, have been reported [8, 10, 11].

3.1 Fundamental Composition of Artificial Neural Networks

Neural networks are algorithms inspired by the way the human brain works. The goal is to learn patterns from data to make decisions or predictions and discoveries, as illustrated in Figure 4 . a general artificial neural network model that contains the main components The basic structure of a neural network, represents the X1.....Xn input , W1.....Wn weights, Y output, and activation functions. It consists of layers (input - hidden - output).

The neural network initially takes raw data, such as images, text, numbers, etc., in the

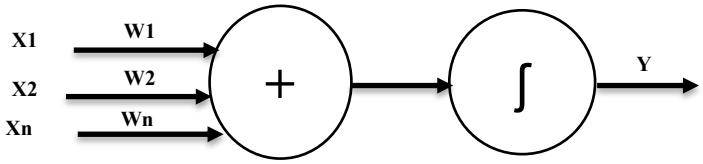

Fig. 4. General artificial neural network model

input layer. The hidden layers are processed and transformed step by step, and each layer learns something new. In the end, the network presents the result, such as image recognition or weather prediction, in the output layer. Each connection within the network has a weight that controls the importance of the information (the values that the inputs are multiplied by to determine their effect on the output). The network decides whether the information is important or not based on the activation functions, which determine whether the node is "active" or "inactive," Without these function, the neural network becomes linear and unable to learn complex patterns. The most famous activation function is the sigmoid function, which converts values between 0 and 1 but may cause the gradient to disappear. The softmax function is used in the last layer of multi-class classification, as illustrated in Figure 5 [13]. Common activation functions in artificial neural networks (NNs).

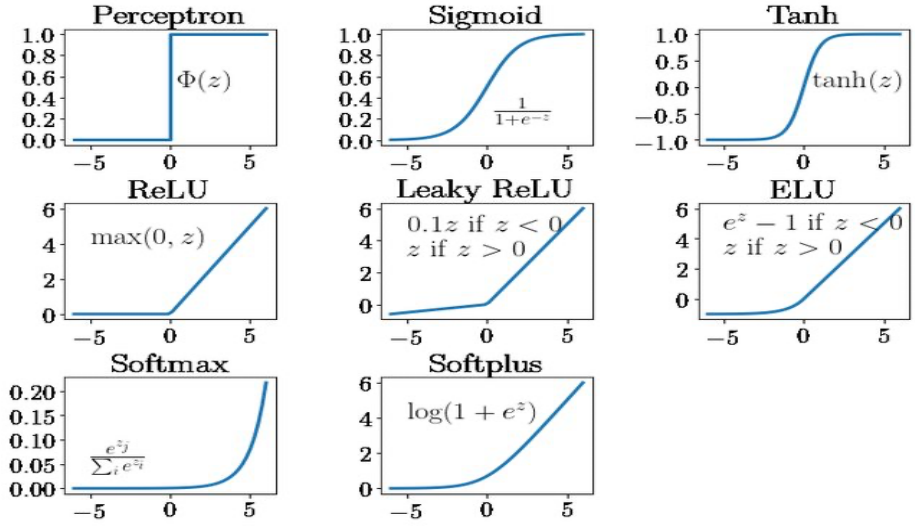

Fig. 5. Activation functions in artificial neural networks.

3.2 Classification of Neural Networks

Neural networks can be classified as shown in the following in table 1.

Table 1. Classification of Neural Networks

Neural networks	Type of learning	Category
Multi-Layer Perceptron	Supervised	Feedforward
Convolutional Neural Network	Supervised	Feedforward
Bidirectional Associative Memory	Supervised	Feedback
Recurrent Neural Network	Unsupervised	Feedback
Self-Organizing Map	Unsupervised	Competitive:
Adaptive Response Theory	Unsupervised	Competitive

Feedforward networks are a type of artificial neural networks in which data moves from an input layer to an output layer in one direction. They may have multiple hidden layers for easier analysis, and are used for different tasks. Feedback networks are among the most complex neural network structures, and CNN is an example of a Feedforward network in artificial neural networks,

Feedback networks consist of an artificial neural network that is capable of learning from information-dense sequential data due to its structure. This makes it a powerful tool. Its memory-preserving, learning, analysis, and prediction properties make it versatile in artificial intelligence applications, especially in recurrent neural networks (RNNs),

Competitive networks are a type of artificial neural networks that learn through the process of competition between neurons to respond to certain inputs, leading to self-organization and effective memory retention of patterns learned over time. Learning in competitive networks is usually unsupervised. Their ability to classify and recognize patterns along with clustering, image processing and data mining makes them powerful tool in various applications. Competitive networks are often associated with Self-Organizing Maps (SOMs) and Adaptive Resonance Theory (ART) [12].

4 Malware Representation as Images

With the ongoing digital transformation in many areas, including cybersecurity, there are increasing problems in detecting, classifying, preventing or, mitigating malware in electronic systems and devices. An emerging malware visualization technique involves converting malware binary files into grayscale (0-255), colored, or binary (0-1) images. To convert them, the malware code or binary files are used to convert them into a two-dimensional array of pixels. As illustrated in Figure 6.

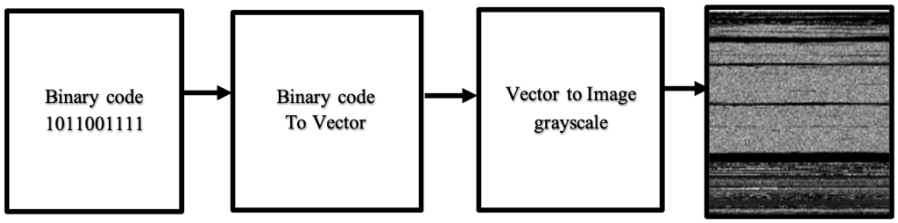

Fig. 6. Representation of Malware as an Image

Researchers have used this visual similarity to detect and classify malware. The malware binary code was visualized as grayscale images. The study visualized the codes of 25 malware families as grayscale images with values ranging from zero to 255 (0: black, 255: white) [14].
Grayscale images contain important features such as structural differences, information density, and visual similarities between malware families, all of which contribute to the effectiveness of the malware detection process. These features enable deep learning models to classify malware accurately and efficiently,
Binary image analysis with deep learning is used. A proposed method uses convolutional neural networks (CNN) to analyze binary images. This method allows the model to learn complex patterns and features from images, which improved the accuracy of malware detection [15].

5 Image-Based Malware Classification

A method, image-based malware classification, applies artificial intelligence models to the information and identification of malware. They are converted from malware files or binary code into a binary image, RGB, or grayscale image. The image dataset is divided into training data and test and verification data. Neural network techniques are trained to classify malware into families or order them with respect to their severity. This can detect malware more accurately and efficiently, outperforming traditional methods that fail to detect some malware, enhancing cybersecurity by integrating detection and response to potential and advanced malware threats.
A malware family is defined as a group of sub-malware programs that share similar attack data. These families have distinct characteristics and behaviors that are similar to each other, which contributes to the smooth detection and classification of malware. The image representation of a particular family is completely difference from that of a different family belonging to another type. This is a result of converting the programming code into a binary file, which is referred to as (visual similarity). If old samples are used to execute new binary files, the resulting binary files will be similar. In most cases, converting an executable file into a grayscale image will help detect and classify the differences between samples belonging to the same family, as illustrated in Figure 7. Two grayscale malware families.

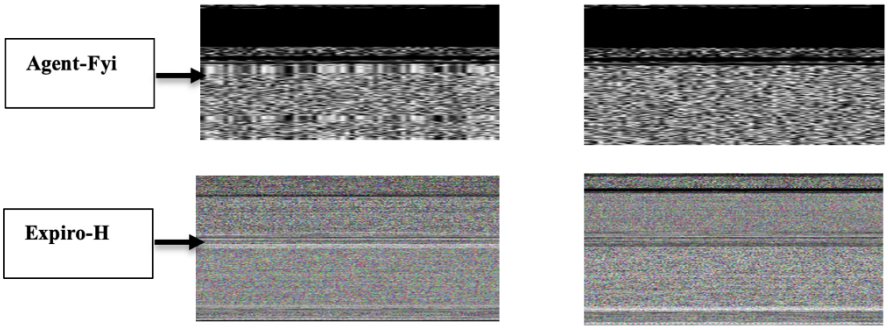

Fig. 7. Grayscale representation of the binary content of two types of malware samples.

6 Malware Analysis and Traditional Malware Detection Techniques

Malware detection, classification, and analysis is critical to cybersecurity and signature-based and behavioral detection methods are traditional techniques that have been used. As malware grows and evolves to become more sophisticated, and the obfuscation techniques used by malware become more sophisticated, these methods are no longer able to protect against undetected infections. These methods can be leveraged through static analysis, which is performed by extracting features from static malware stored on disk, and hybrid methods that combine dynamic and static analysis.

6.1 Malware Analysis

Malware analysis in cybersecurity has been of great importance to understanding and mitigating the risks posed by malware. By using various analysis techniques and tools, analysts can examine and understand malware to better determine its behavior, intent, and potential impact on systems, networks, and modern malware sites.

6.1.1 Static Analysis

Static malware analysis is a traditional malware detection technique used to detect malware without running it. This type of detection analyses source code to extract information and understand malware behaviour. At employs techniques such as string analysis, file structure examination, and syntax analysis. Static analysis can provide initial information on malware and often yields limited and simplistic insights that are not sufficient to identify the malware accurately.
Static analysis works as the starting point in the malware analysis process, also in deciding whether such indicators of compromise need to be located and analysed immediately. The approaches of static analysis include delving into the code and the internal

processes of a binary file. It is necessary to know the operating system and the programming language for deeper insight. Static analysis works to quickly analyse a multitude of files at once; however, malware in its intense variants slows down its execution to complex malware so as not to be detected; its limitations bedevil its effectiveness in the most sophisticated threats in the evasion category [16].
Static analysis is complicated by binary obfuscation techniques that transform malware binaries to resist reverse engineering. This makes static analysis expensive and less reliable, especially against sophisticated evasion techniques used by malware developers. Behavioural analysis techniques and machine learning are crucial for understanding malware and identifying new variants [17].
The static analysis of malware has certain drawbacks. Malware constructors utilise certain binary code obfuscation approaches to cause great difficulties in reverse engineering, making it utterly expensive and very error-prone. Critical information about the size of data structures or variables can especially be overlooked during the static analysis stage, raising more complexities towards the static analysis stage [18].

6.1.2 Dynamic Analysis

Dynamic malware analysis examines a sample of malware as it runs on the system. In simulated or controlled environments, such as virtual machines, emulators, simulators, and sandboxes, it is possible to analyze and monitor malicious code. By analyzing malware's interactions with the system and network, its changes, and interactions with other processes on the host system, we can understand how it affects a host system.
Dynamic analysis does not require the disassembly of the executable and reveals the true behaviour of the malware, which is often resistant to static analysis. Malware can behave differently in a simulated environment than in a real-world setting, making detection more challenging. Certain malware behaviours are also conditional, making them difficult to identify in a controlled setting [18].
Advanced malware can use alternative behaviour in a virtual environment in order to mitigate the notice of its presence and thus become difficult to identify [17]. Dynamic analysis faces challenges such as evasion techniques, limited data availability, and computational complexities. Additionally, it emphasises the need for robust systems that can handle advanced threats, address limitations of current methods, and provide early detection capabilities [18].

6.2 Malware Traditional Detection Techniques

Traditional malware detection techniques include signature-based, behavior-based, or hybrid detection, each with their own strengths and weaknesses. Signature-based detection is effective for known threats but struggles with new variants, while behavior-based detection provides broader coverage but can lead to false positives and user frustration.

6.2.1 Signatures –based and Behavioral –based Detection Malware

Detection using signatures can be effective for the recognition of known threats, but it only works based on certain predefined patterns. This method, however, becomes ineffective when used against never-before-seen or altered malware that does not conform to existing signature patterns. Such detection methods are also ineffective against polymorphic and code-morphing malware, as they can easily evade detection by altering their signatures. Although the behaviour-based approach is capable of recognising a wider spectrum of malware, including unknown ones, it is also prone to false positives due to their over inclusiveness. In addition to that, such methods can also mistakenly identify non-malicious executable as malware if their behaviour contextually aligns with malware behaviour [19].

A new approach blending signature and behaviour detection was proposed in a study. This system does have some efficiency against certain known and emerging malware threats but cannot combat new styles of complex malware that do not have defined signatures as such, which can leave gaps in security. The system also performs real-time network packet analysis, which is very demanding and resource-intensive. This may cause performance problems depending on the amount of data traffic since the system has to work hard to keep up with the data flow. In addition, this system tries to minimise false positives, but the very nature of behavior-based detection is bound to cause some misclassification [20].

7 Related Work

Models for malware detection and classification using supervised and unsupervised neural networks, as well as other types of artificial neural networks (ANNs), are being created and enhanced in a various way for use in malware detection and classification techniques.

7.1 Models Supervised

In study [21], Multilayer Perceptron (MLP) achieved superior performance compared to other traditional malware detection techniques on a specific ransomware dataset. However, the study used a single type of malware, which limits the detection of other types of high-quality datasets and hinders the effectiveness of the proposed methods.

Other study research in [22], “Android Malware Detection Using Backpropagation Neural Network”, the model classifies apps as malicious or benign. Experimental results indicate that the proposed method can discriminate with 100% accuracy. However, the study used small datasets, which exposes the risk of overfitting. In addition, the absence of cross-validation or testing phase metrics (such as precision, recall, or F1 score) weakens the reliability of asserting 100% training accuracy. While the simplicity of the model using lightweight features and a shallow network may favor deployment on resource-limited devices, this trade-off may weaken the detection effectiveness in complex threat environments.

An earlier study was conducted [23], proposes a malware classification methodology that integrates dynamic behavioural profiling with a backpropagation (BP) neural network. This approach achieves 86% classification efficiency in detecting real-world malware samples. Although, the authors present the challenges in feature selection, computational overhead, and variability across malware classes. The study did not use precise evaluation metrics. In addition, some malware families exhibit significantly lower classification performance than others.

In a study [24], it improved detection of persistent malware with an accuracy of 97.8%, but its reliance on imported functions and imbalanced data reflects broader challenges in this field. The study also used a binary neural network classifier to classify Windows Portable Executable (PE) files, making the method fail to detect other or advanced threats in other environments. Furthermore, the data was not validated on diverse and recent datasets (such as EMBER and VirusShare).

The study [25], APSO-CNN-SE: An Adaptive Convolutional Neural Network Approach for IoT Intrusion Detection aimed to develop an efficient and effective intrusion detection system (IDS) specifically designed for Internet of Things (IoT) networks. The proposed model APSO-CNN-SE shows significant improvements in detection accuracy compared to the baseline CNN model and outperforms other models. However, it cannot solve the problem of imbalanced data distribution and is limited to IoT networks.

Another study [26], proposes an image-based malware classification model using convolutional neural networks, EfficientNet. The proposed model, EfficientNetB1, achieved 99% accuracy in malware classification. Compared to other models, while EfficientNetB1 outperforms heavier models, the study neglects comparisons with lightweight architectures (such as MobileNet and ShuffleNet) specifically designed for efficiency. However, resizing/padding can delete or distort critical byte sequences, unlike API-based vectors that preserve subtle feature semantics. The use of an outdated dataset raises concerns about information accuracy, generalisability, and resilience against modern, sophisticated threats.

In another study [27], convolutional neural networks were used to detect malware by converting binary malware files into greyscale images, with an accuracy of 90% in distinguishing between benign and malicious files. However, the error rate was high, at 14.02%, which leads to misclassification of benign files as malicious. Adversarial attacks (such as malware pixel jamming) can also exploit CNN vulnerabilities. Their limited adaptability to sophisticated threats underscores the need for advanced frameworks.

According to [28], the study focused on image-based malware detection using convolutional neural networks (CNN) and CRNN networks. The results show that the CRNN outperformed a traditional CNN, achieving 92.24% accuracy, 93.12% precision, and 92.56% F1 score, while the basic CNN scored ~67% across metrics. Nevertheless, noise/obfuscation (e.g., Gaussian noise) reduced CRNN accuracy to 86.67%. Additionally, those evasion tactics could undermine CNNs' reliability by disrupting learnt patterns.

In another study [29], proposed a CNN-based method for Android malware detection. The LeNet-E model (entropy-color images) achieved 98.5% accuracy on ARM and 99.0% on x86 datasets, outperforming SVM (88.0% on x86) and Logistic Regression

(96.1%). Regardless, the study did not evaluate scalability, computational demands, or applicability to other platforms (e.g., Windows, IoT). Furthermore, LeNet/AlexNet are outdated; modern architectures (e.g., MobileNet, EfficientNet) could improve accuracy while maintaining efficiency.

In addition, study [30], proposed a CNN-based ransomware detection model by analyzing Portable Executable (PE) headers. The model achieved 93.33% and 95.11% accuracy on two test sets, with faster training/testing times compared to prior methods. Although, reliance on static PE header analysis limits detection of packed/encoded ransomware. Moreover, attackers can trivially modify PE headers (e.g., section names, timestamps) to evade detection, a vulnerability not tested in the study.

In research conducted by [31], introduced a web-based malware detection system using a 1D-CNN to classify Portable Executable (PE) files as malicious or benign. The model achieved high accuracy across three datasets: 98.85% (Benign and Malicious PE Files), 98.37% (Classification of Malware), and 97.25% (MalwareDataSet). However, the reliance on header-based features limits detection of sophisticated threats, particularly those targeting fragmented IoT ecosystems or employing signature evasion tactics.

In an earlier study [32], the VBDN framework, an image-based CNN algorithm for multi-class malware detection. Evaluated on four public datasets, VBDN achieved over 90% accuracy, surpassing traditional machine learning classifiers. Albeit, the framework struggles with adaptability with "obscured" malware variants, indicating gaps in handling poorly defined or novel threat types.

Other study research [33], introduced IMCFN, a fine-tuned convolutional neural network (CNN) for malware classification using the Malimg dataset (9,435 greyscale/cooler images across 25 families). The model achieved 98.82% accuracy with color images versus 98.27% for greyscale, demonstrating that cooler enhances feature discrimination. High precision (98.85%) and recall (98.81%) suggest strong generalisation across malware families. After all, attackers could manipulate cooler channels (e.g., altering pixel RGB values) to deceive the model, a vulnerability not addressed.

In research, conducted [34], worked a Malware Classification Framework comparing two approaches: a dense neural network (DNN) applied to binary files and a convolutional neural network (CNN) for malware image classification. Using the 2015 Microsoft Malware Classification Challenge dataset, the CNN-based method achieved 97.8% accuracy, outperforming the DNN. While demonstrating the efficacy of image-based deep learning, the framework struggles with sophisticated evasive malware and risks overfitting due to reliance on static signatures.

In addition, research [35], an improved CNN model for malware classification using greyscale images generated from binary files. The Malimg dataset (9,339 images across 25 families) was used, with binaries converted to 2D arrays. The custom CNN achieved 98.03% accuracy, outperforming pretrained models like VGG16 (96.96%) and ResNet50 (97.11%). A hybrid CNN-SVM (linear kernel) reached 99.59% accuracy. However, computational inefficiencies in feature extraction (e.g., GIST) and scalability on larger datasets remain challenges. Additionally, absent FPR/FNR data obscures operational risks (e.g., false positives).

7.2 Models Unsupervised

An earlier study [36], used a self-organising map (SOM) for unsupervised clustering analysis of malware behaviour, analyzing 270,000 samples to address limitations of traditional antivirus (AV) classification. The SOM generated behaviour-based clusters that outperformed AV vendor classifications in accuracy. Although reliance on inconsistent AV labels for validation introduced potential biases. Furthermore, malware can mimic benign behaviours or randomise API call sequences to evade detection.

In addition to the study [37], a hybrid malware classification framework combining self-organising feature maps (SOFMs), logistic regression, and Bayesian networks. Using continuous machine activity data (e.g., execution behaviours), the model reduced overfitting and improved accuracy by 7.24–25.68% over traditional methods like Random Forest. However, performance significantly dropped on unseen datasets, indicating that the model may encounter difficulties in accurately classifying new cases and revealing generalisation gaps.

In another study [38], developed an explainable intrusion detection system (X-IDS) using self-organising maps (SOMs) to balance interpretability and accuracy. Evaluated on NSL-KDD (91% accuracy) and CIC-IDS-2017 (80% accuracy), the model enabled transparent threat clustering but suffered from overfitting on the latter dataset. Despite high accuracy, the lack of precision, recall, and F1-score metrics limits a holistic performance assessment.

In study, research [39], developed an LSTM-based model to classify five malware types (backdoors, rootkits, Trojans, viruses, and worms) and benign software using a custom MC-dataset-multiclass (19,740 samples). The model achieved 67.6% overall accuracy, with a high true positive rate (TPR) for rootkits (92.19%) but struggled with Trojans (51.06%). Despite balanced malware/benign samples, performance variability across classes highlights challenges in generalisation.

Another research study [40], enhanced ransomware detection by integrating an Attended Recent Inputs (ARI) cell into LSTM networks. Using a dataset of 12,500 Windows-emulated ransomware/benign file sequences, the ARI-LSTM achieved 93% accuracy, outperforming the standard LSTM (87%). Attention mechanisms improved detection of local behavioural patterns (e.g., repeated encryption calls), validated by ROC curves showing low false positives. Moreover, the model's generalisability to other malware types and real-world efficiency remain unaddressed.

The research study [41], proposes an LSTM- and GRU-based RNN model for malware classification using API call sequences extracted via dynamic analysis (Cuckoo Sandbox, Alkanet Tracer). The model classified eight malware types, achieving strong validation accuracy despite variable sequence lengths. A softmax output layer generated malware family probabilities. However, the study omitted overfitting risks and real-world deployment challenges (e.g., adversarial noise, computational constraints). Additionally, dynamic analysis (Cuckoo Sandbox) is slower and resource-heavy compared to other methods, limiting scalability.

In another research study [42], developed LSTM and bidirectional RNN models to detect malware in cloud environments using 40,680 live cloud samples (malicious/be-

nign). The models analysed resource metrics (CPU/memory usage) to distinguish behaviours, achieving >99% accuracy. LSTMs trained faster than bidirectional RNNs, with input process sequencing critical to performance. Although the study assumed single-malware compromises per VM (unrealistic for multi-tenant clouds) and relied on average resource patterns, leaving stealthier malware undetected. Furthermore, accuracy inflation: >99% accuracy on controlled live samples may not reflect real-world multi-tenant chaos or adversarial attacks.

In another study [43], developed an RNN-based model for Android malware detection, achieving 98.58% accuracy. While effective at distinguishing malware from benign apps, RNNs may fail against apps mimicking benign sequences or injecting noise (e.g., junk API calls). Additionally, the need for continuous model updates to address evolving threats.

In study [44], they developed a novel method, a hybrid Convolutional Gated-Recurrent-Unit (CGRU) model for malicious URL detection, combining CNNs (spatial feature extraction) and GRUs (temporal sequence processing). Trained on 405,000 URLs (65k benign, 340k malicious), the model achieved 99.6% accuracy, outperforming manual feature-based methods and standalone neural networks. However, adaptability to evolving threats via real-time learning remains unaddressed.

Another study [45], used an RNN model to predict malware within the first 5 seconds of execution using initial behavioural data (e.g., API calls, registry changes). Trained on a dataset of benign files, APTs, and ransomware from VirusShare, the model achieved 94% accuracy, enabling early threat intervention. Moreover, maintaining efficacy requires frequent retraining with new malware samples, necessitating automated update pipelines.

An earlier study conducted a study [46], designed an Adaptive Resonance Theory (ART-2)-based intrusion detection system (IDS) for local area networks (LANs). Using the KDD'99 dataset, the hybrid system (data acquisition + ART-2) achieved a 98% recognition rate for attacks and 96% for normal traffic. Although, it struggled with rare attack types (e.g., multihop, guess_passwd, buffer_overflow), which comprised only 0.003% of the dataset, leading to high false negatives. Furthermore, synthetic datasets and outdated traffic lack modern attack vectors (e.g., IoT exploits, ransomware).

In another study[47], introduced a hierarchical ART-2m neural network for malware detection, combining adaptive resonance theory (ART) with Control Flow Graph (CFG) vectorisation via Graph2Vec. Analysed 500 executables using API call sequences and CFG structural patterns, achieving high speed/accuracy in detecting known and unknown malware variants. Outperformed naive Bayes classifiers in adaptability and efficiency but faced scalability and generalisation challenges due to limited data. Additionally, malware can alter CFGs (e.g., dead code insertion, control flow flattening) to evade Graph2Vec's structural analysis.

In addition, another study [48], used adaptive resonance theory in a hybrid intrusion detection system (IDS) combining Projective Adaptive Resonance Theory (PART) and K-means clustering to enhance network security. Evaluated on the KDD'99 dataset, the hybrid model reduced training time while maintaining detection accuracy. The approach improved system performance through multi-directional feature optimisation but remained confined to traditional network security frameworks, lacking validation

on modern threats. Furthermore, it is unspecified how PART and K-means interact—e.g., sequential vs. parallel processing—limiting reproducibility.

Earlier research [49], developed a database intrusion detection system (DIDS) using Adaptive Resonance Theory (ART) integrated with data mining techniques. By analyzing database access logs and query patterns, the model demonstrated adaptability to dynamic environments, achieving high accuracy and low false positive rates for both common and rare intrusion types. However, scalability and applicability to modern, complex database architectures (e.g., distributed/NoSQL systems) remain unaddressed.

Other research [50], explored the application of Adaptive Resonance Theory (ART) variants (fuzzy ART, ART2-A, PCA-MART2) to enhance intrusion detection systems (IDS). Key findings include the importance of optimal parameter tuning (vigilance, learning rate) and the benefits of hybridising ART with methods like PCA. While ART techniques show promise for real-time IDS, the study calls for further refinement to address scalability and complex threat landscapes.

In addition, research [51], explored the use of Adaptive Resonance Theory 1 (ART1) to enhance intrusion detection systems (IDS). The model dynamically learns new intrusion patterns without overwriting prior knowledge. Fuzzy ART further improved IDS performance by enabling anomaly detection for unknown threats. However, the study lacked empirical benchmarks against state-of-the-art methods, limiting validation of its claims. Additionally, attackers can manipulate binary features (e.g., flipping protocol flags) to mimic benign patterns.

7.3 Hybrid Models

An earlier study [52], proposed hybrid CNN-LSTM models for Android malware detection. The LSTM-CNN architecture achieved 98.53% accuracy, outperforming standalone models: MLP (94.73%), CNN (87.91%), LSTM (95.90%), and CNN-LSTM (96.76%). However, the model's generalisability to other malware types/datasets is uncertain, and performance may degrade with parameter changes or evolving threats.

Other study research [53], evaluated Multi-Layer Perceptron (MLP), Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN) for ransomware classification. The MLP achieved 100% binary classification accuracy, outperforming CNN (94%) and RNN (79%). Moreover, the dataset is small size (notably an inconsistent split: 372 training and 93 testing) and imbalance (3:1 malware-to-benign ratio) raised concerns about overfitting and generalisability. Regardless, MLP is superior performance but the need for larger datasets to validate robustness.

Another study [54], researched a hybrid CNN-BiLSTM model for malware detection and classification. The system achieved 99.44% detection accuracy (data length: 1,200) and 95.4% classification accuracy (data length: 2,000), with a false positive rate (FPR) of 0.23%, demonstrating strong performance in identifying malware while minimising false alarms. However, ambiguities in dataset composition and testing protocols raise concerns about reproducibility and other environments' applicability. Furthermore, while 0.23% FPR is low, unstated false negative rates (FNR) could mask critical missed detections (e.g., ransomware).

8 Review Methodology

This paper adopted a systematic literature review to survey advances in malware detection and classification using artificial neural networks (ANNs). The steps are as follows:

8.1 Research Objectives and Questions

The primary objectives of this review are to:
1. Identify and categorise the various ANN architectures applied in malware detection and classification.
2. Compare between Supervised and Unsupervised Learning models via studies.
3. Highlight the challenges and future directions in ANN-based malware analysis.

The review addresses the following research questions:
1. RQ1: Which ANN architectures are most prevalent and effective for malware detection and classification?
2. RQ2: How do different ANN-based approaches comparing to models supervised and unsupervised?
3. RQ3: What gaps exist in current literature, and what are the potential avenues for future research?

8.2 Study Field Area

Focused on malware detection/classification in diverse environments: web-based infrastructure, PE headers, cloud platforms, mobile (Android/Windows), IoT devices, URLs, API-call logs, databases, and network traffic.
Examine four primary ANN paradigms: Convolutional Neural Networks (CNNs), Self-Organising Maps (SOMs), Recurrent Neural Networks (RNNs), and Adaptive Resonance Theory (ART) networks and other models while also noting emerging and hybrid architectures.

8.3 Study and Analysis

Assessing Supervised vs. Unsupervised vs. Hybrid. Comparing the strengths, limitations, and applicability of each paradigm.
Environment-specific Insights identify which ANN techniques excel in particular settings (e.g., CNNs for image-based detection, SOMs for clustering unknown samples).
Identifying gaps, highlighting areas that need further research, such as generalising malware that need to be detected, dealing with imbalanced datasets, and reducing computational overhead.
By this transparent and reproducible methodology, the review provides a comprehensive, up-to-date mapping of how ANN techniques are being leveraged and where they fall short in combating the evolving malware landscape.

Table 2. Comparative between Supervised and Unsupervised Learning models

Aspect	Supervised Models	Unsupervised Models
Data Requirement	Require large, accurately labeled datasets (e.g., malware family tags) to train classifiers.	Operate on unlabeled data ideal when labels are scarce or expensive to obtain.
Typical Architectures	CNNs (e.g., EfficientNetB1, Inception), MLPs, RNNs (LSTM/GRU)	Self-Organizing Maps (SOM), Adaptive Resonance Theory (ART) networks
Detection Accuracy	Very high—often >95 % (many CNNs reach 99 %)	Good but generally lower SOMs around 91 %, ART up to 98 % on common attacks, but rare events drop sharply
Generalization	Can generalize well if trained on diverse, balanced labels; vulnerable to zero-day variants without labeled examples.	Naturally, group's novel behaviors, but clusters may misclassify rare or evolving malware families.
Interpretability	Often “black-box”—hard to explain why a sample is flagged.	Clusters and prototypes (e.g., in SOM/ART) offer visual/semantic insights into malware behavior.
Computational Cost	High, especially deep CNNs and hybrid models (e.g. CNN–BiLSTM).	Moderate to low: SOM/ART training is typically faster and memory-efficient.
Adaptivity	Static once trained—requires retraining on new labels.	Adaptive by design (ART’s stability-plasticity) and can incorporate new patterns without full retraining.
False Positives	Can achieve very low FPRs (down to 0.2 % with CNN–BiLSTM) but sensitive to adversarial noise.	FPR varies ART can maintain low FPRs on common attacks but struggles with uncommon patterns.
Best Use Cases	When high-quality labeled datasets exist and accuracy is paramount.	When labels are unavailable or for exploratory analysis of emerging threats.

9 Results and analysis

9.1 Supervised Models

MLPs are among the earliest neural network models applied in supervised malware classification. In [21], an MLP classifier trained on a Kaggle ransomware dataset outperformed traditional techniques in accuracy and precision, although its effectiveness

was limited by dataset quality and availability. Similarly, the model in [22], which focused on Android malware detection using MLPs and features like APK size and battery usage, achieved 100% accuracy during the training phase. However, the small sample size significantly undermined its generalizability to real-world threats. In [23], a behavioral-based MLP approach attained 86% accuracy for malware samples and 99% for benign files using dynamic features from the HABO system, confirming MLPs' potential in behavioral analysis.

CNNs have demonstrated exceptional performance in image-based malware detection due to their ability to automatically extract spatial features from visual representations of malware binaries. In [26], the EfficientNetB1 CNN model, trained on the Microsoft Malware Classification Challenge (MMCC) dataset, achieved 99% accuracy with significantly reduced computational time (0.1881 sec), outperforming other CNN models like ResNet and Inception. Another study [27] converted malware binaries into images and trained a CNN based on the Inception V3 architecture, reaching over 90% accuracy. However, challenges like false positives and the evolving nature of malware persisted. In [28], a CNN and a hybrid CRNN model were evaluated using the Malicia dataset. While both models performed well, the CRNN achieved 92.24% accuracy, with transfer learning further enhancing its performance. However, noise and obfuscation in malware images reduced detection accuracy, illustrating the vulnerability of CNNs to adversarial modifications. Similarly, [29] utilized CNNs (LeNet and AlexNet) to classify Android malware images derived from Hilbert space-filling curves and entropy visualizations. Detection rates ranged between 98.5% and 99%, highlighting the method's effectiveness, though scalability and computational cost remained issues.

The study in [30] used CNNs to analyze Portable Executable (PE) headers and achieved accuracy rates between 93.33% and 95.11% for detecting ransomware, demonstrating fast processing times. Meanwhile, a one-dimensional CNN model in [31] attained up to 98.85% accuracy on various datasets, validating the efficiency of lightweight CNN architectures in malware detection.

Other Supervised Architectures. In [35], a CNN model combined with Support Vector Machine (SVM) achieved an outstanding accuracy of 99.59% on the Malimg dataset. This hybridization demonstrated that CNNs, when paired with classical machine learning algorithms, could yield highly accurate and robust classifiers. Furthermore, [33] showed that CNN performance improved when malware images were represented in color rather than grayscale, achieving 98.82% accuracy compared to 98.27% for grayscale, indicating the importance of input feature representation.

Another study [34] compared Dense Neural Networks (DNNs) with CNNs for malware image classification and found CNNs significantly more effective, achieving 97.8% accuracy. However, the reliance on specific image patterns made the model susceptible to evasion techniques. Lastly, [32] introduced a general-purpose CNN-based framework (VBDN) that maintained accuracy above 90% across multiple datasets and outperformed traditional machine learning classifiers.

9.2 Unsupervised Models

SOMs are unsupervised neural networks capable of transforming complex, high-dimensional data into lower-dimensional representations, making them useful for clustering and visualization. In [36], a clustering analysis based on SOM effectively grouped over 270,000 malware samples into behavior-based clusters, outperforming traditional antivirus classification methods. Another study [37] applied Self-Organizing Feature Maps (SOFMs) on continuous machine activity data, demonstrating improved classification accuracy (a 7.24% to 25.68% increase over traditional methods). Despite these successes, the performance of SOMs diminished significantly when applied to previously unseen datasets, highlighting generalization challenges. Additionally, SOM-based X-IDS systems [38] achieved high accuracy rates (91% with NSL-KDD and 80% with CIC-IDS-2017 datasets) and offered interpretable results, though overfitting and the lack of comprehensive evaluation metrics remain concerns.

Although RNNs are predominantly applied in supervised learning, several studies have leveraged their capacity to analyze sequential patterns without explicit labels. LSTM models used in [39] achieved 67.6% accuracy across a multi-class malware dataset, showing promise in identifying temporal malware patterns. However, they struggled with low true positive rates for certain malware categories (e.g., Trojans). Another study [40] introduced an Attended Recent Inputs (ARI-LSTM) model, which improved ransomware detection accuracy to 93%, surpassing standard LSTM's 87%. These results suggest that integrating attention mechanisms enhances pattern recognition in malware behavior. Nevertheless, the cross-generalizability of such models to diverse malware families was not thoroughly evaluated.

ART models offer a unique balance between stability (retaining learned knowledge) and plasticity (adapting to new inputs), making them highly suitable for the dynamic nature of malware threats. The ART2-based model in [46] reported a 98% recognition rate for attacks and 96% for normal traffic using the KDD'99 dataset. However, its performance was limited when classifying rare attacks (e.g., buffer overflow, guess_passwd), which comprised a small portion of the dataset. Additional studies [46], [48], and [49] demonstrated that ART networks are capable of fast and accurate detection, particularly in database and network intrusion contexts. Notably, [49] highlighted that ART outperforms SOM and radial basis function networks in detection speed and adaptability, although comparisons with newer deep learning techniques were lacking.

9.3 Hybrid Models

In [52], researchers proposed hybrid models by combining CNN with LSTM to enhance Android malware detection using the Drebin dataset (129,013 samples). The LSTM-CNN architecture outperformed standalone models, achieving an accuracy of 98.53%, compared to 95.90% with LSTM, 87.91% with CNN, and 94.73% with MLP. This indicates that the combination effectively captures both spatial and sequential features. However, the study noted that the performance might not generalize to other types of malware or datasets due to overfitting and dependence on sequential input representation like Bag-of-Words (Bow).

The study in [54] introduced a CNN-BiLSTM model using image-based representations of malware. This hybrid model achieved 99.44% accuracy on malware detection for sequence lengths of 1200 and 95.4% accuracy for classification with a sequence length of 2000, with a very low false positive rate of 0.23%. The CNN module effectively extracts spatial features from malware images, while BiLSTM captures bidirectional dependencies in data, enhancing classification accuracy. Despite the promising results, performance is sensitive to data length, and improper tuning can lead to accuracy degradation.

A comparative study in [53] evaluated the performance of MLP, CNN, and RNN on a dataset of 4,000 ransomware samples (1,000 benign and 3,000 malicious). The MLP achieved 100% accuracy in binary classification, while CNN and RNN reached 94% and 79% respectively. Although MLP performed best in this scenario, the small dataset size limits generalization. This also suggests that simple architectures can outperform complex models if feature engineering is well executed, but for nuanced, real-world data, deeper hybrid networks are more scalable.

10 Discussion

Supervised ANN models, especially CNNs and MLPs, have shown excellent performance in malware detection tasks. Most CNN-based studies achieved >95% accuracy, with some models like EfficientNetB1 and CNN+SVM reaching up to 99.59%. Converting malware binaries into images (grayscale, RGB, entropy-based) has proven highly effective for CNN training. Model performance often hinges on dataset size, balance, and feature diversity. Smaller or imbalanced datasets can lead to overfitting and reduced generalizability. Combining CNNs with classical classifiers (e.g., SVM) or integrating attention mechanisms improves performance. However, computational demands for training deep models. Vulnerability to adversarial inputs (e.g., obfuscation, noise). Limited performance on unseen or highly polymorphic malware.

Unsupervised neural networks, particularly SOMs and ART, offer interpretability and flexibility in handling evolving malware threats. Their capability to cluster unknown malware and detect anomalies is valuable in early-stage threat identification. However, generalization challenges performance often degrades on unseen or rare samples. Dependence on data representation effectiveness varies significantly based on input features, such as machine activity logs or image-based encodings. Overfitting risks especially in complex or noisy datasets like CIC-IDS-2017. Evaluation gaps some studies lacked standard performance metrics (e.g., F1-score, precision) and did not benchmark against state-of-the-art supervised methods. Despite these challenges, the ability of unsupervised models to adapt to new malware behaviors without complete retraining is a notable advantage. ART networks, in particular, show great potential due to their stability-plasticity trade-off and robustness in intrusion detection settings.

Hybrid neural networks clearly offer superior performance in malware detection and classification tasks by integrating the strengths of multiple architectures. CNN-BiLSTM and LSTM-CNN models reached 98–99.4% accuracy, outperforming many

single-model approaches. CNNs capture spatial features from malware images or encoded sequences, while LSTMs handle temporal behavior patterns effectively. The CNN-BiLSTM model, in particular, maintained a low 0.23% FPR, making it suitable for practical deployment. However, hybrid models are computationally expensive and may not be suitable for low-resource environments. Performance is sensitive to sequence length, image size, and dataset balance. Most studies used custom or limited datasets (e.g., Drebin, PE files), raising concerns about real-world applicability and robustness against threats.

11 Conclusion

The rapid evolution of malware, characterised by increasingly sophisticated and evasive techniques, has rendered traditional detection methods such as signature-based analysis and behavioural analysis ineffective when used against malware that does not conform to such stored signature patterns, new types of complex malware that do not have specific signatures, and polymorphic, variable code malware. Static analysis is limited by the widespread use of obfuscation. Dynamic analysis fails with counter-analysis techniques and is computationally expensive. These methods are insufficient as malware authors use advanced techniques to hide the malicious intent of the program, making it difficult to analyze them. Hence, artificial neural networks (ANNs) have started to gain the attention of researchers in the field of malware detection and classification, especially in malware image classification and analysis. This study emphasises the potential of artificial neural networks (ANNs) in addressing the challenges of traditional methods and providing high performance, adaptability, and scalability in malware detection and classification.

Supervised models, such as CNNs have shown exceptional performance in image-based malware classification, achieving high accuracy. However, limitations such as information loss during image resizing and exposure to adversarial noise remain critical concerns. Other networks, like MLP networks, have shown high accuracy in binary classification tasks such as for malware detection. However, their reliance on fixed features limits their effectiveness against obfuscated malware.

Unsupervised models, such as Self-organising Maps (SOMs) and Adaptable Resonance Theory (ART) Networks, have demonstrated some interpretable and adaptable solutions to non-stationary data with high malware detection accuracy by the unsupervised models. However, they have become unreliable due to low dataset classifications and sporadic attack patterns. Although RNN/LSTM-based methods have been used to analyses time series data, such as API calls, it has proven challenging to generalise the model across different malware families and frequent retraining is required.

Hybrid model approaches, such as LSTM-CNN, have outperformed standalone networks with high accuracy on a dataset. Additionally, image-based methods using grey-scale/color representations have benefitted from the pattern recognition strengths of CNNs but struggled to deal with the computational overhead and diversity of datasets. Moreover, hybrid models, CNN-BiLSTM, have shown improved detection rates with

high accuracy by combining sequential and spatial analysis, although their effectiveness is highly dependent on the quality and size of the data. Finally, more research is needed on artificial neural networks for malware detection and classification, as they can reduce the need for traditional methods. In the future, researchers may consider developing more efficient, scalable, and adaptive neural network-based methods to address the changing nature of malware.

References

1. "World Wide Attacks - Live." Accessed: Jun. 28, 2025. [Online]. Available: https://attackmap.sonicwall.com/live-attack-map/
2. Z. Zhao, D. Zhao, S. Yang, and L. Xu, "Image-Based Malware Classification Method with the AlexNet Convolutional Neural Network Model," Secur. Commun. Netw., vol. 2023, pp. 1–15, Apr. 2023.
3. E. K. Kabanga and C. H. Kim, "Malware Images Classification Using Convolutional Neural Network," J. Comput. Commun., vol. 6, no. 1, Art. no. 1, Dec. 2017.
4. D. Gibert, C. Mateu, and J. Planes, "The rise of machine learning for detection and classification of malware: Research developments, trends and challenges," J. Netw. Comput. Appl., vol. 153, p. 102526, Mar. 2020.
5. Y. Harayama et al., "Artificial Intelligence and the Future of Work," in Reflections on Artificial Intelligence for Humanity, B. Braunschweig and M. Ghallab, Eds., Cham: Springer International Publishing, 2021, pp. 53–67.
6. J. E. (Hans). Korteling, G. C. Van De Boer-Visschedijk, R. A. M. Blankendaal, R. C. Boonekamp, and A. R. Eikelboom, "Human- versus Artificial Intelligence," Front. Artif. Intell., vol. 4, p. 622364, Mar. 2021.
7. T. Thomas, A. P. Vijayaraghavan, and S. Emmanuel, Machine Learning Approaches in Cyber Security Analytics. Singapore: Springer Singapore, 2020.
8. M. Pawlicki, R. Kozik, and M. Choraś, "A survey on neural networks for (cyber-) security and (cyber-) security of neural networks," Neurocomputing, vol. 500, pp. 1075–1087, Aug. 2022.
9. K. Barik, S. Misra, K. Konar, L. Fernandez-Sanz, and M. Koyuncu, "Cybersecurity Deep: Approaches, Attacks Dataset, and Comparative Study," Appl. Artif. Intell., vol. 36, no. 1, p. 2055399, Dec. 2022.
10. A.-M. Ghimeş and V.-V. Patriciu, "Neural network models in big data analytics and cyber security," in 2017 9th International conference on electronics, computers and artificial intelligence (ECAI), IEEE, 2017, pp. 1–6.
11. B. R. Maddireddy and B. R. Maddireddy, "Neural Network Architectures in Cybersecurity: Optimizing Anomaly Detection and Prevention," vol. 01, no. 02, 2024.
12. O. A. Montesinos López, A. Montesinos López, and J. Crossa, "Fundamentals of Artificial Neural Networks and Deep Learning," in Multivariate Statistical Machine Learning Methods for Genomic Prediction, Cham: Springer International Publishing, 2022, pp. 379–425.
13. N. Johnson et al., Machine Learning for Materials Developments in Metals Additive Manufacturing. 2020.
14. L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: visualization and automatic classification," in Proceedings of the 8th International Symposium on Visualization for Cyber Security, Pittsburgh Pennsylvania USA: ACM, Jul. 2011, pp. 1–7.

15. V. Patil, S. Shetty, A. Tawte, and S. Wathare, “Deep Learning and Binary Representational Image Approach for Malware Detection,” in 2023 International Conference on Power, Instrumentation, Control and Computing (PICC), IEEE, 2023, pp. 1–7.
16. H. A. Noman, Q. Al-Maatouk, and S. A. Noman, “A Static Analysis Tool for Malware Detection,” in 2021 International Conference on Data Analytics for Business and Industry (ICDABI), Sakheer, Bahrain: IEEE, Oct. 2021, pp. 661–665.
17. R. Yadav and D. Singh, “Malware Detection and Analysis Tools,” Int. J. Recent Innov. Trends Comput. Commun., vol. 11, no. 11s, pp. 735–744, 2023.
18. J. Smallman, “A Survey on Malware Detection and Analysis,” J. Sci. Technol., vol. 5, no. 4, pp. 1–14, Jul. 2024.
19. A. K. Chakravarty, A. Raj, S. Paul, and S. Apoorva, “A study of signature-based and behaviour-based malware detection approaches,” Int J Adv Res Ideas Innov Technol, vol. 5, no. 3, pp. 1509–1511, 2019.
20. D. L. S. Punyasiri, “Signature & Behavior Based Malware Detection,” 2023.
21. K. S, A. S, S. S, A. M, and K. M, “Malware Detection Using Neural Network,” Int. J. Innov. Res. Eng., pp. 31–35, May 2023.
22. F. Al Huda, W. Firdaus Mahmudy, and H. Tolle, “Android Malware Detection Using Backpropagation Neural Network,” Indones. J. Electr. Eng. Comput. Sci., vol. 4, no. 1, p. 240, Oct. 2016.
23. Z.-P. Pan, C. Feng, and C.-J. Tang, “Malware Classification Based on the Behavior Analysis and Back Propagation Neural Network,” ITM Web Conf., vol. 7, p. 02001, 2016.
24. B. Bashari Rad, M. Shahpasand, and M. Nejad, “Malware classification and detection using artificial neural network,” J. Eng. Sci. Technol., vol. 13, pp. 14–23, Jul. 2018.
25. Y. Ban, D. Zhang, Q. He, and Q. Shen, “APSO-CNN-SE: An Adaptive Convolutional Neural Network Approach for IoT Intrusion Detection,” Comput. Mater. Contin., vol. 81, no. 1, pp. 567–601, 2024.
26. R. Chaganti, V. Ravi, and T. D. Pham, “Image-based malware representation approach with EfficientNet convolutional neural networks for effective malware classification,” J. Inf. Secur. Appl., vol. 69, p. 103306, Sep. 2022.
27. C.-M. Chen, S.-H. Wang, D.-W. Wen, G.-H. Lai, and M.-K. Sun, “Applying Convolutional Neural Network for Malware Detection,” in 2019 IEEE 10th International Conference on Awareness Science and Technology (iCAST), Morioka, Japan: IEEE, Oct. 2019, pp. 1–5. Accessed: Feb. 13, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/8923568/
28. B. Palomino, “Image-Based Malware Detection using Convolutional Neural Network Techniques,” Master of Science in Data Science, San Jose State University, San Jose, CA, USA, 2023.
29. N. Lachtar, D. Ibdah, and A. Bacha, “Toward Mobile Malware Detection Through Convolutional Neural Networks,” IEEE Embed. Syst. Lett., vol. 13, no. 3, pp. 134–137, Sep. 2021.
30. F. Manavi and A. Hamzeh, “Ransomware Detection Based on PE Header Using Convolutional Neural Networks.,” ISeCure, vol. 14, no. 2, 2022.
31. A. Alqahtani, S. Azzony, L. Alsharafi, and M. Alaseri, “Web-Based Malware Detection System Using Convolutional Neural Network,” Digital, vol. 3, no. 3, pp. 273–285, Sep. 2023.
32. Y. Liu, H. Fan, J. Zhao, J. Zhang, and X. Yin, “Efficient and Generalized Image-Based CNN Algorithm for Multi-Class Malware Detection,” IEEE Access, vol. 12, pp. 104317–104332, 2024.

33. D. Vasan, M. Alazab, S. Wassan, H. Naeem, B. Safaei, and Q. Zheng, "IMCFN: Image-based malware classification using fine-tuned convolutional neural network architecture," Comput. Netw., vol. 171, p. 107138, Apr. 2020.
34. M. Khan, D. Baig, U. S. Khan, and A. Karim, "Malware Classification Framework using Convolutional Neural Network," in 2020 International Conference on Cyber Warfare and Security (ICCWS), Islamabad, Pakistan: IEEE, Oct. 2020, pp. 1–7.
35. S. S. Lad and A. C. Adamuthe, "Malware classification with improved convolutional neural network model," Int. J. Comput. Netw. Inf. Secur., vol. 9, no. 6, p. 30, 2020.
36. R.-S. Pirscoveanu, M. Stevanovic, and J. M. Pedersen, "Clustering analysis of malware behavior using Self Organizing Map," in 2016 International Conference On Cyber Situational Awareness, Data Analytics And Assessment (CyberSA), Jun. 2016, pp. 1–6.
37. P. Burnap, R. French, F. Turner, and K. Jones, "Malware classification using self organising feature maps and machine activity data," Comput. Secur., vol. 73, pp. 399–410, Mar. 2018.
38. J. Ables et al., "Creating an Explainable Intrusion Detection System Using Self Organizing Maps," Jul. 15, 2022, arXiv: arXiv:2207.07465.
39. E. D. O. Andrade, J. Viterbo, C. N. Vasconcelos, J. Guérin, and F. C. Bernardini, "A Model Based on LSTM Neural Networks to Identify Five Different Types of Malware," Procedia Comput. Sci., vol. 159, pp. 182–191, 2019.
40. R. Agrawal, J. W. Stokes, K. Selvaraj, and M. Marinescu, "Attention in Recurrent Neural Networks for Ransomware Detection," in ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, United Kingdom: IEEE, May 2019, pp. 3222–3226.
41. C. Li and J. Zheng, "API Call-Based Malware Classification Using Recurrent Neural Networks," J. Cyber Secur. Mobil., May 2021, doi: 10.13052/jcsm2245-1439.1036.
42. J. C. Kimmel, A. D. Mcdole, M. Abdelsalam, M. Gupta, and R. Sandhu, "Recurrent Neural Networks Based Online Behavioural Malware Detection Techniques for Cloud Infrastructure," IEEE Access, vol. 9, pp. 68066–68080, 2021.
43. M. Almahmoud, D. Alzu'bi, and Q. Yaseen, "ReDroidDet: Android Malware Detection Based on Recurrent Neural Network," Procedia Comput. Sci., vol. 184, pp. 841–846, 2021.
44. W. Yang, W. Zuo, and B. Cui, "Detecting Malicious URLs via a Keyword-Based Convolutional Gated-Recurrent-Unit Neural Network," IEEE Access, vol. 7, pp. 29891–29900, 2019.
45. M. Rhode, P. Burnap, and K. Jones, "Early-stage malware prediction using recurrent neural networks," Comput. Secur., vol. 77, pp. 578–594, Aug. 2018.
46. D. G. Bukhanov and V. M. Polyakov, "Detection of network attacks based on adaptive resonance theory," J. Phys. Conf. Ser., vol. 1015, p. 042007, May 2018.
47. D. G. Bukhanov, V. M. Polyakov, and M. A. Redkina, "Detection of malware using an artificial neural network based on adaptive resonant theory," Prikl. Diskretn. Mat., no. 2, pp. 69–82, 2021.
48. P. Tiwari, P. Mishra, U. Singh, and R. Itare, "New Adaptive Resonance Theory Based Intrusion Detection System," in Second International Conference on Computer Networks and Communication Technologies, S. Smys, T. Senjyu, and P. Lafata, Eds., Cham: Springer International Publishing, 2020, pp. 745–754.
49. A. Brahma and S. Panigrahi, "Database Intrusion Detection Using Adaptive Resonance Network Theory Model," in Computational Intelligence in Data Mining, H. S. Behera, J. Nayak, B. Naik, and D. Pelusi, Eds., Singapore: Springer, 2020, pp. 243–250.

50. K. Champaneria, B. Shah, and K. J. Panchal, “Survey of Adaptive Resonance Theory Techniques in IDS,” 2014.
51. A. Saxena and A. Sharma, “Intrusion Detection System to Improve the Detection Rate using ART-1 Algorithm,” vol. 2, no. 2, 2015.
52. M. A. Halim, A. Abdullah, and K. A. Z. Ariffin, “Recurrent neural network for malware detection,” Int J Adv. Soft Compu Appl, vol. 11, no. 1, pp. 43–63, 2019.
53. H. Madani, N. Ouerdi, A. Boumesaoud, and A. Azizi, “Classification of ransomware using different types of neural networks,” Sci. Rep., vol. 12, no. 1, p. 4770, Mar. 2022.
54. H. Kim and M. Kim, “Malware Detection and Classification System Based on CNN-BiLSTM,” Electronics, vol. 13, no. 13, p. 2539, Jun. 2024.

الكشف عن البرامج الضارة وتصنيفها باستخدام الشبكات العصبية الاصطناعية: مراجعة

محمد ابو سعيدة[1] ، محمود منصور[2]

المعهد العالي للعلوم والتقنية القره بوللي[1]

m.abosaeeda@uot.edu.ly

كلية تقنية المعلومات، جامعة طرابلس، طرابلس، ليبيا[2]

mah.mansour@uot.edu.ly

الملخص: أدى التطور السريع للبرمجيات الخبيثة، وخاصةً المتغيرات متعددة الأشكال والمتحولة، إلى تدني فعالية أساليب الكشف التقليدية، مثل الكشف القائم على التوقيع والكشف السلوكي. هدفت هذه الدراسة إلى مراجعة شاملة للشبكات العصبية الاصطناعية (ANNs) للكشف عن البرمجيات الخبيثة وتصنيفها، وذلك من خلال مراجعة شاملة لنماذج الشبكات العصبية الأكثر استخداماً. ركزت الدراسة على النماذج الخاضعة للإشراف، والنماذج غير الخاضعة للإشراف، والنماذج الهجينة في بيئات متنوعة.

تشير نتائج الدراسة إلى أن النماذج الخاضعة للإشراف تحقق دقة استثنائية (أكثر من 95%)؛ بينما توفر النماذج غير الخاضعة للإشراف قابلية للتفسير والتكيف مع التهديدات المتطورة، ولكنها تواجه تحديات في التعميم على البيانات غير المرئية. في المقابل، تجمع النماذج الهجينة بين استخراج السمات المكانية والزمانية، محققةً دقة تصل إلى 99.4%، وإن كانت تكاليفها الحسابية أعلى. تؤكد هذه الدراسة على أهمية وجود أطر عمل قوية ضد التعتيم، وهياكل فعّالة للبيئات محدودة الموارد، وتعميم مُحسّن عبر عائلات البرمجيات الخبيثة.

الكلمات المفتاحية: اكتشاف البرمجيات الخبيثة، تصنيف البرمجيات الخبيثة، صورة البرمجيات الخبيثة، خوارزميات الشبكات العصبية الاصطناعية.