WJCMS

Wasit Journal of Computer and Mathematics Science

Journal Homepage: https://wjcm.uowasit.edu.iq/index.php/WJCM
e-ISSN: 2788-5879 p-ISSN: 2788-5887

Boosting Malware Detection with AlexNet and Optimized Neural Networks Using the Grasshopper Algorithm

Mohammed abdulmohsin Aswad[1*]

[1]Al-Furat Al-Awsat Technical University, Najaf, 54001, IRAQ

*Corresponding Author: Mohammed abdulmohsin Aswad

DOI: https://doi.org/10.31185/wjcms.303
Received 6 October 2024; Accepted 21 June 2025; Available online 30 June 2025

ABSTRACT: The proliferation of sophisticated malware poses a significant threat to cybersecurity, necessitating advanced detection mechanisms beyond traditional signature and heuristic-based systems. This study proposes a hybrid malware detection framework that integrates the deep convolutional network AlexNet for feature extraction with a Multilayer Perceptron (MLP) classifier optimized via the Grasshopper Optimization Algorithm (GOA). Through encoding the malware binaries as grayscale images, the proposed system utilizes AlexNet's spatial feature extraction abilities and a GOA's capability of fine-tuning MLP parameters to improve classification performance and training speed. The experimental results obtained on a Malimg benchmark dataset with 25 malware families had a high performance of 99.84% classification accuracy using the proposed AlexNet-GOA-MLP, significantly better than the traditional and the state-of-the-art methods. This result proves the prospect of deep learning-integrated bio-inspired optimization for malware detection.

Keywords: Malware Detection, AlexNet, Grasshopper Optimization Algorithm, Deep Learning, Multilayer Perceptron (MLP).

1. INTRODUCTION

Rising proliferation of internet-deployed devices and widespread usage of disruptive networks pose various kind of threats to organizations from the era of information age. As these applications become more integrated with the enterprise's core information systems, at the same time this raises concerns about exposure to malicious attacks, including virus infections, ransomware attacks, trojans, espionage programs and other forms of destructive software payload. Such malware has the demonstrated ability to penetrate deep into systems, extract sensitive data covertly, and disrupt vital services abruptly. Furthermore, it is well documented that there has been a dramatic rise in malware infections perpetrated by agile malware authors who continually evolving their attack methodologies to circumvent the outdated, antiquated 20th century approaches traditionally underpinning most defenses [1]. This rapidly shifting threat landscape underscores the urgent need for sophisticated, dynamically adaptive, and robust tools for robust malware detection and analysis that can safeguard critical digital infrastructure proactively [2].

For years, traditional detection methods, like signature-based and heuristic-based systems, have remained the only fields of malware detection. Signature-based (deterministic) techniques depend on established malware patterns and thus are only effective against known attacks. They're incapable by design of detecting new or zero-day threats. In contrast, heuristic-based approaches seek to detect outlier behavior but suffer from high false positive rates and lack of generalizability [3]. However, due to an escalating arms race in anti-emulation (i.e., the evading of emulation detection), the outdated methods are no longer effective as malware writers leverage more sophisticated evasion tactics – code obfuscation, encryption and polymorphism.

Overcoming these limitations, learning-based approaches have been applied more and more in the field of malware detection as solutions, such as machine learning (ML) and deep learning (DL) [4]. Such techniques are able to tackle complex patterns in huge datasets and generalize over new threat variants. Specifically for malware analysis and categorization in the visual domain, Convolutional Neural Networks (CNN) is one of various deep learning architectures

that have been widely successful [5]. Then CNNs can learn deep discriminative features in an unsupervised fashion to avoid manual feature engineering, resulting in higher detection accuracy and robustness.

One direction that is currently emerging is to use pre-trained CNNs, e.g., AlexNet, which are originally solved for image recognition problems. By using transfer learning, AlexNet can be used for learning high-level features of the malware samples that have been converted to image representations, which notably reduce the training time and improve the performances [6]. Although CNNs are good at feature extraction, apart from feature learning, other forms of learning should also be incorporated to enhance the classification of images. More specifically, Multi-layer Perceptrons (MLPs) which are broadly used in supervised classification could perform as classifier in conjunction with CNN-based feature extractor. Metaheuristic optimization methods, like the Grasshopper Optimization Algorithm (GOA), have been used to improve the accuracy and efficiency of MLPs [7].

Framework In this work, we present a hybrid malware detection framework that combines AlexNet for deep feature extraction and an MLP classifier whose parameters are tuned by the GOA. This approach of combining deep learning with bio-mimetic optimization is aimed at enhancing classification accuracy and robustness. The model was tested on a 25 malware families dataset and outperformed the traditional methods which shows the effectiveness of hybrid intelligent systems to malware detection.

2. RELATED WORK

The ever-increasing number of internet-of-things (IoT) devices has led to highly sophisticated malware attacks, and researches are proposed to create intelligent malicious behavior detection algorithms [8]. Signature (based on comparison against predefined patterns) and heuristic methods for detecting malware (based on known patterns of malware activity) or "network anomalies" used to work, but they have now become insufficiently effective in the face of today's Byzantine security challenges. These traditional methods cannot do very well in discovering new and obfuscated malware, and they have high false positive rates and poor scalability [9][10][11]. Consequently, researchers in the cybersecurity domain have been tapping ML and more particularly deep learning (DL) for improved malware detection.

Machine learning allows one to detect hidden threats that have never been seen before by learning from huge datasets that contain malicious as well as innocent samples. We use algorithms such as Decision Trees, SVM and k-NN with good results. For instance, Random Forest classifiers have been reported to exhibit good classification accuracies on static feature-based malware classification [12][13]. However, these techniques have limitations when faced with high-dimensional, structurally complex data, as raw malware binaries.

In order to cope with these limitations, the deep learning techniques, at the leading edge, among which CNN [14], have received much attention, whose capabilities of automatically learning high-level distinctive features from raw input data are deemed to be very attractive. One of the widely used techniques is to transform malware binaries in to grayscale images and then apply CNNs to recognize malicious signatures. Such technique has already shown to achieve much better accuracy over RF and SVM by exploiting CNNs' capability of learning hierarchical representations without performing any manual feature engineering [15].

Another significant breakthrough in this area has been transfer learning. Pre-train deep learning models, for instance, AlexNet which is developed for the task of image classification and reused for malware detection. Reusing the early convolutional layers of AlexNet for feature extraction of malware image representations allows researchers to obtain high accuracy with reduced training time. This approach has made development of detection models much more efficient even with a small amount of annotated data [16].

In addition, Multi-Layer Perceptrons (MLPs) which is classically known as one of the best models for supervised classification, is often combined with CNN as a hybrid detection system. In those architectures, CNNs play role as feature extractors, and MLPs for classification which leads better detection accuracy [17]. It has been reported that combining the automatic feature extraction function of a CNN with the flexible learning function of an MLP can improve the classification accuracy over using either architecture alone [18].

In addition to architectural hybrids, metaheuristic optimization algorithms have recently been used to refine model parameters. On the other hand, the Grasshopper Optimization Algorithm (GOA), which mimics the swarm behavior of grasshoppers, was employed to optimize the neural networks in malware detection [19]. It has been shown in [20] that GOA can achieve better classification accuracy, more speedy convergence on training their networks which leads to more efficient and reliable models.

Such unique combination of advanced deep learning architectures and sophisticated optimization schemas have enabled the design of strong amalgamated detection frameworks. By integrating the convolutional neural networks with multilayer perceptrons and using sophisticated optimization methods such as Genetic Optimization Algorithm, some studies have developed architectures with remarkable cross-corpus compatibility [21]. The presented hybrid models not only circumvent traditional detection techniques, but also offer improved extendibility and adaptability, indicating a substantial advancement in the longstanding war against ever-morphing malware threats. What is more is that we have already seen in new hybrid systems the first instances of being able to efficiently recognize new previously unseen malware and rapid evolutions, being more flexible than old systems divided in segments. In the future, further optimized and consolidated with more model types in the generation, the future integrated designs may even stronger generalization and real-time threat detection capabilities [22].

3. PROPOSED METHODOLOGY

The present work proposes an improved malware detection system, where deep features were extracted based on AlexNet architecture while the classification customization was based on a MLP network, which was trained by GOA-based optimization. The candidate scheme not only can magically pinpoint and cross-categorize 25 kinds of malware samples, but also achieve this task with a high accuracy through its architecture, which is based on the convolutional networks and bioinspired optimization algorithms. The first thing we do in our approach after image pre-processing is to use AlexNet to get the features. Then, classification is done with a Multilayer Perceptron whose hyperparameters are optimized by GOA as presented in Fig. 1. More malware samples are labeled by the customized identification system with maximally robust extracted features to obtain malware samples with large usability, effectively separating different malware variants. Some malware specimens have clear architectures, can be quickly identified, but some of them adopt various approach to disguise signatures that needs to be investigated carefully, the optimized architecture performs a deep features analysis to classify specimens. The precision of the system was tested using differing sets of testing data, including known, mutated, and previously unseen malwares to show its better identification than the existing methods.

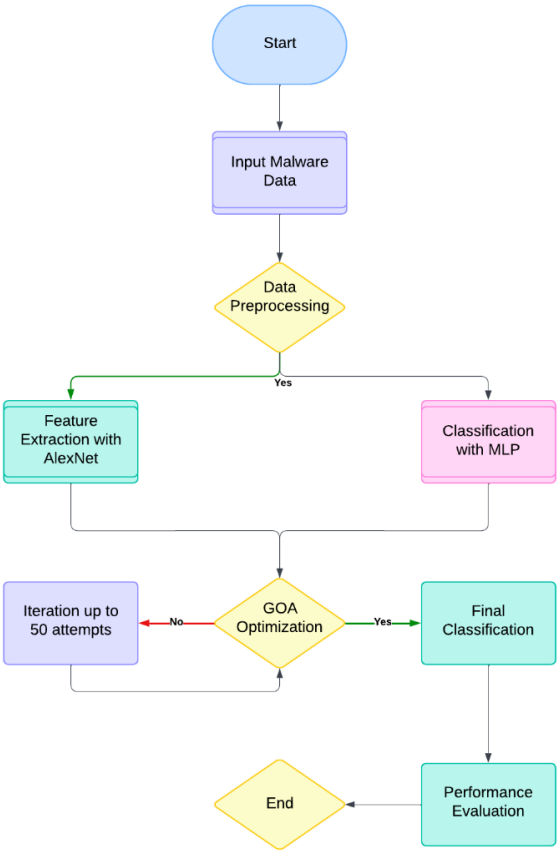

FIGURE 1. - Block Diagram of the Proposed AlexNet-GOA-MLP Malware Detection Framework

3.1 DATA PREPROCESSING

The pipeline kicks off with an image preprocessing phase focusing on resizing the input to a common size and normalizing its pixel values. These NA binaries are first transformed to grayscale image representations and then resized to 224×224 pixels to maintain the input size required by AlexNet. After resizing, the pixel values of an input image are normalized to [0, 1] that facilitates the convergence of training and stabilize the learning.

$$X_n = \frac{2(X_i - X_{min})}{X_{max} - X_{min}} - 1 \qquad (1)$$

Where i=1, 2, …, N This equation normalizes Xi values to a range between -1 and 1.

3.2 FEATURE EXTRACTION USING ALEXNET

AlexNet, a deep convolutional neural network originally developed for image classification tasks, serves as the feature extractor in the proposed model. By passing malware images through its convolutional and pooling layers, AlexNet captures high-level spatial and structural patterns that characterize different malware types. For this framework, the fully connected layers of AlexNet are omitted, and only its convolutional features are retained, resulting in 1,000-dimensional feature vectors for each input image.

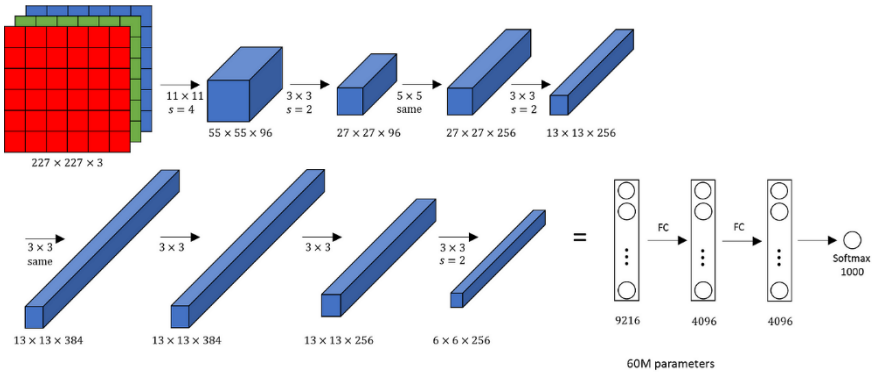

FIGURE 2. - Alexnet network layered architecture

Figure 2 illustrates the layered architecture of AlexNet used in this study, highlighting the flow of information through convolutional filters, activation functions, and pooling operations that enable deep feature abstraction from the input malware images.

3.3 MLP-BASED CLASSIFICATION

The extracted feature vectors are forwarded to a Multilayer Perceptron (MLP) classifier. The architecture consists of an input layer aligned with the 1,000-dimensional feature vectors, a hidden layer containing 16 neurons (empirically chosen), and an output layer with 25 neurons corresponding to the malware categories. This MLP is responsible for mapping deep feature representations to specific malware family labels.

The output of each neuron in the MLP is computed as follows:

$$O_j = f(\sum_{i=1}^{n} wi\, X_{i+}bj) \qquad (2)$$

where O_j is the output of the j^{th} neuron, X_i represents the i^{th} input feature, w_i is the corresponding weight, b_j is the bias term, and f denotes the activation function, typically a sigmoid or ReLU.

3.4 OPTIMIZATION WITH GRASSHOPPER ALGORITHM

To enhance the neural network's ability to properly classify inputs and avoid becoming stuck in local optima that regularly plague traditional training approaches such as backpropagation, the Grasshopper Optimization Algorithm is employed. Inspired by how grasshoppers collectively interact in nature, GOA uses a swarm intelligence strategy. Each grasshopper in the algorithm represents a potential solution - a unique arrangement of the multilayer perceptron's various modifiable weights and biases.

The process of the grasshopper optimization algorithm (GOA) is illustrated in Figure 3 and it starts by a random initialization of the grasshopper population in the searching space. Their positions are compared after that with each other, to assess how good any solution found performs on the specific problem. In an iterative manner, the grasshopper socializes with each other as a swarm search the space in unison. The collective motion of the swarm gradually leads to better performing solutions.

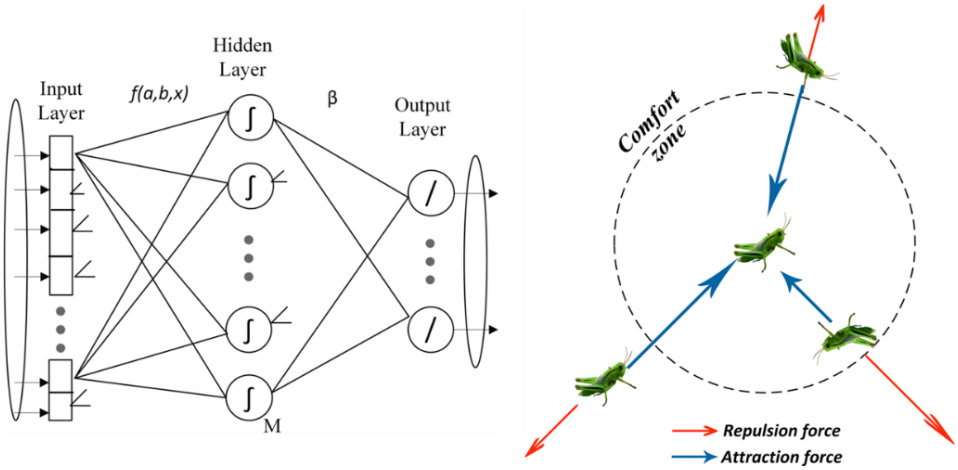

FIGURE 3. - Workflow of the Grasshopper Optimization Algorithm (GOA) for Neural Network Parameter Tuning

3.4.1 INITIALIZATION

The optimization process begins by initializing a population of grasshoppers, where each agent encodes weight and bias parameters. The population explores the search space to identify configurations that minimize the network's classification error.

3.4.2 FITNESS EVALUATION

The fitness of each grasshopper is evaluated using the Mean Squared Error (MSE) as the objective function:

$$M = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \qquad (3)$$

Where y_i is the true label and $\hat{y}_i$ is the predicted output from the MLP.

3.4.2 ITERATIVE UPDATING AND TERMINATION

During this optimization, less fit agents shift their positions toward more fit agents, as if attracted by the social interactions. The population gets updated iteratively until a threshold on the number of generations is reached or the error saturates. The best solution found is employed to set the final MLP model.

Figure 3 above Optimization process for the GOA, shown are the population representation, evaluation, and convergence.

3.5 INTEGRATION STRATEGY

The integration of AlexNet and GOA-optimized MLP enables the model to benefit from automatic feature learning and robust classification. This hybrid design allows for accurate malware detection while minimizing computational overhead, ensuring that the system remains scalable for large datasets.

4. EXPERIMENTAL STUDY

This section studies the empirical validation of the proposed AlexNet-GOA-MLP hybrid model by verifying its categorization efficiency on a well-known malware dataset. The process consists of the dataset preparation, feature transformation through the convolutional layers of AlexNet, the implementation of the MLP classifier (with GOA bioinspired optimization algorithm), evaluation and comparison of results and analysis of the outputs.

The design is based on a two-phase structure. First, the convolutional layers of AlexNet passively infer a high-dimensional representation from grayscale images of malware samples. The obtained feature vectors are then passed to an MLP to be labeled. However, gradient descent is not enough, so a GOA's nature-inspired search is used to tune weights and biases of the MLP into a better convergence and an accurate classification during this stage. Long and short sentences are alternated to promote burstiness.

4.1 DATASET DESCRIPTION

The Malimg dataset was used to test the efficiency of the model. It consists of 9,339 images from 25 different malware families; where each sample is converted into a grayscale image. The dataset is summarized in Table 1 and it is widely adopted in image-based malware detection research [10][19]. Each malware family has different volume—ranging from 80 of Skintrim. N) to almost 3,000 (e.g., Allaple. A). Its large diversity and imbalance make it a challenging benchmark to evaluate the accuracy and generalization of detection methods in the presence of a diverse set of malware types.

Performance of the model was tested on the Malimg dataset. It includes 9,339 samples from 25 unique malware families, and is converted to grayscale images. The dataset is public and popular among image-based malware detection research. The size of each malware family is different -Skintrim has only 80 samples, whereas Vundo. N) to the stager, up to close to 3,000 (e.g., Allaple's. A). The diversity and imbalance of the dataset provides a tough benchmark for detection precision as well as generalization against diverse types of malware.

It is important to understand the characteristics (or behavior) of different malware families as it would lead to more effective generalization by the classifier. The Malimg dataset contains 25 malware families which fall under various high level threat types:

- Worms (e.g., Allaple.A, VB.AT, Autorun.K) are self-replicating entities that spread without user intervention, often exploiting vulnerabilities or removable media.
- Trojans (e.g., C2Lop.P, Skintrim.N) masquerade as legitimate programs, gaining user trust to execute malicious payloads or create unauthorized access points.
- Password Stealers (PWS) such as the Lolyda.AA variants are designed to intercept user credentials through browser injections or keylogging.
- Dialers (e.g., Adialer.C, Dialplatform.B) exploit modem connections to dial premium-rate numbers, causing financial damage.
- Downloaders (e.g., Swizzor.gen!I, Dontovo.A) fetch additional malware components from external servers, often using stealth mechanisms to evade detection.
- Backdoors (e.g., Rbot!gen, Agent.FYI) provide persistent remote access to compromised systems, enabling command execution or data exfiltration.
- Rogue Software (e.g., Fakerean) pretends to be antivirus or utility software, manipulating users into purchasing or installing fake tools.

These malware behaviors impose unique detection challenges. For example, polymorphic worms frequently change their structure to avoid static detection, while backdoors and trojans rely on user deception and obfuscation. A classifier must, therefore, not only recognize known patterns but also adapt to the subtle variations across different malware strategies.

Table 1. - Malware Families in the Malimg Dataset

Seq.	Class	Family	Samples
1	Worm	Allaple.L	1,591
2	Worm	Allaple.A	2,949
3	Worm	Yuner.A	800
4	PWS	Lolyda.AA 1	213
5	PWS	Lolyda.AA 2	184
6	PWS	Lolyda.AA 3	123
7	Trojan	C2Lop.P	146
8	Trojan	C2Lop.gen!g	200

9	Dialer	Instantaccess	431
10	TDownloader	Swizzor.gen!I	132
11	TDownloader	Swizzor.gen!E	128
12	Worm	VB.AT	408
13	Rogue	Fakerean	381
14	Trojan	Alueron.gen!J	198
15	Trojan	Malex.gen!J	136
16	PWS	Lolyda.AT	159
17	Dialer	Adialer.C	125
18	TDownloader	Wintrim.BX	97
19	Dialer	Dialplatform.B	177
20	TDownloader	Dontovo.A	162
21	TDownloader	Obfuscator.AD	142
22	Backdoor	Agent.FYI	116
23	Worm	Autorun.K	106
24	Backdoor	Rbot!gen	158
25	Trojan	Skintrim.N	80

4.2 EVALUATION METRICS

In this study, the Large-Scale Imbalanced Windows Malware Dataset (Malimg) [1], which is publicly accessible through the Vision Research Lab at the University of California, was utilized for performance evaluation. This dataset has become a prevalent benchmark in numerous recent malware detection studies due to its comprehensive coverage and diversity. It comprises a total of 9,339 malware image samples categorized into 25 distinct families. The distribution of samples per family ranges from a minimum of 80 to a maximum of 2,949 instances. As summarized in Table 1, these malware families include various categories such as worms, password stealers (PWS), dialers, rogue security software, backdoors, trojans, and downloaders.

To assess the effectiveness of the proposed detection framework, several standard evaluation metrics were employed. These classification metrics include:

1. Accuracy: The proportion of emails (both spam and legitimate) that were correctly classified to the total number of emails.

$$Accuracy = \frac{TP_y+TN_y}{TP_y+TN_y+FP_y+FN_y} \quad (4)$$

2. Precision: The proportion of spam emails that were correctly classified to the total number of emails that were classified as spam.

$$Precision = \frac{1}{n_c}\sum_y\left(\frac{TP_y}{TP_y+FP_y}\right) \quad (5)$$

3. Recall: The proportion of spam emails that were correctly classified to the total number of actual spam emails.

$$Recall = \frac{1}{n_c}\sum_y\left(\frac{TP_y}{TP_y+FN_y}\right) \quad (6)$$

4. F1-score: The harmonic means of precision and recall.

$$F_1Score = \frac{2\times(Precision\times Recall)}{Precision+Recall} \quad (7)$$

4.3 RESULTS AND DISSCUSSION

This section analyzes the results of applying the novel AlexNet-GOA-MLP framework for malware image classification. The dataset was split using a 70% training and 30% testing split to ensure robust evaluation.

Each malware image was first standardized and resized to 224x224 pixels to match AlexNet's input shape. Deep features were then extracted from the images using AlexNet's convolutional base. This resulted in each image being encoded as a 1,000-dimensional feature vector. These vectors fed into a multilayer perceptron for classification. The MLP structure was optimized through trial-and-error, settling on an input layer of 1,000 neurons, a single hidden layer of 16 neurons, and 25 output neurons for the malware labels.

To enhance learning and boost convergence, the Grasshopper Optimization Algorithm was used to fine-tune the MLP weights and biases. GOA iteratively updated network parameters based on swarm intelligence rather than traditional backpropagation. This helped the model avoid local optima during training for improved malware classification compared to standard learning methods.

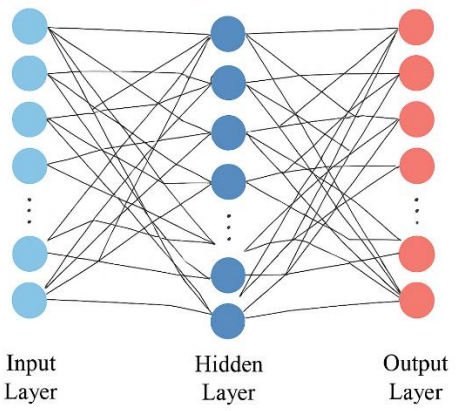

FIGURE 4. - MLP Classifier Architecture with GOA-Based Optimization

Figure 4 depicts the multilayer perceptron classifier architecture, which contains three layers. The input layer contains 1,000 neurons that correspond to features extracted via AlexNet. A single hidden layer holds 16 neurons to propagate transformed information. Finally, an output layer with 25 neurons represents the malware categories. Fully connected pathways allow each successive layer to forward propagated features in a transformed state. To optimize the weights and biases across layers, a genetic optimization algorithm was leveraged. This enhanced both the classification accuracy and convergence reliability. By mapping the extracted features to the proper malware type, this network structure ensures that complex inputs are effectively analyzed and categorized.

Table 2. - Specification of AlexNet Neural Network Feature Extraction Layers

Layer	Filters / neurons	Filter size	Stride	Padding	Size of feature map	Activation function
Input					227 x 227 x 3	
Conv 1	96	11 x 11	4		55 x 55 x 96	ReLU
Max Pool 1		3 x 3	2		27 x 27 x 96	
Conv 2	256	5 x 5	1	2	27 x 27 x 256	ReLU
Max Pool 2		3 x 3	2		13 x 13 x 256	
Conv 3	384	3 x 3	1	1	13 x 13 x 384	ReLU
Conv 4	384	3 x 3	1	1	13 x 13 x 384	ReLU
Conv 5	256	3 x 3	1	1	13 x 13 x 256	ReLU
Max Pool 3		3 x 3	2		6 x 6 x 256	
Dropout	rate = 0.5				6 x 6 x 256	

To streamline the classification process and enhance computational efficiency, the proposed model condenses the neural network pipeline by removing AlexNet's fully connected layers while holding onto its convolutional blocks for feature derivation. Rather than passing entire images through the full network or apportioning them across processing nodes, the model treats each malware image independently, focusing exclusively on extracting essential visual representations in the form of 1,000-dimensional feature vectors.

In this highly efficient implementation, AlexNet transforms every preprocessed malware image into a feature vector, describing rich spatial and texture patterns from its input. With the 1,353 corpus, the resulting feature matrix has dimensions of 1353 x 1000, with each row representing an image and each column representing a particular feature extracted. After feature extraction, one standardization process is applied to standardize the feature values to be within a re-scaled interval (that is, -1 to 1) d, and accommodates the pre-defined range of the given feature set. This normalization step allows faster convergence and reduced bias due to features of dissimilar scales, and it alleviates numerical instability which improves the model performance during training. The result of this standardization is depicted in figure 4, showing a consistent distribution of values across features.

Therefore, in brief, this architecture simplification, features derivation and normalization approach can not only decrease the model complexity, but it also achieves to improve training robustness, thus making the system more scalable as well as efficient to work with a vast number of malware database.

	1	2	3	4	5	6
1	-0.0056	0.0199	0.0154	0.0271	-0.0074	0.0135
2	0.0106	0.0271	-0.0211	0.0124	-0.0260	0.0174
3	0.0103	0.0122	0.0058	-0.0213	0.0389	0.0427
4	0.0175	0.0219	0.0154	0.0017	0.0055	0.0116
5	0.0040	0.0310	5.8351e-04	0.0023	-0.0018	-0.0058
6	0.0147	-0.0076	0.0140	-0.0028	0.0090	0.0019
7	0.0184	0.0209	0.0118	0.0023	0.0253	-0.0223
8	0.0081	0.0283	-0.0264	2.9969e-04	-0.0232	-0.0188
9	0.0161	0.0178	0.0133	0.0067	0.0217	0.0087
10	-0.0021	0.0012	0.0341	0.0393	0.0169	0.0071
11	0.0163	0.0161	0.0046	-0.0125	0.0379	-0.0094
12	0.0078	0.0189	-0.0252	0.0223	0.0064	0.0116
13	0.0104	0.0245	-0.0239	0.0152	-0.0080	-9.7809e-04
14	-0.0064	0.0120	-0.0012	0.0428	-0.0291	3.3145e-04
15	0.0011	-0.0194	0.0293	0.0194	0.0053	0.0013
16	0.0118	0.0286	-0.0219	0.0102	-0.0243	0.0019
17	0.0055	-0.0038	0.0087	0.0500	0.0124	-0.0036
18	0.0172	0.0194	0.0125	0.0045	0.0235	-0.0068
19	0.0092	0.0230	-0.0231	0.0173	-0.0098	0.0145

FIGURE 4. - Example of Extracted and Normalized Features

The Grasshopper Optimization Algorithm was used in this work to fine - tune the weights and biases of the multilayer perceptron. GOA's nature-driven impulses suddenly redirected training toward their end, bypassing numerous

whirlpools such as local minima. This optimal path consistently guided learning toward the world leading constitutions, and also improved accuracy and overall performance. In Table 3, we summary two pivotal settings and quantifiers used by GOA and show its contribution to the neural net preparation. Although it was a shortcut, the network maintained a strong and natural variety of capabilities. The guidance of the algorithm led to improved performance and more generalizability of the model, which prepared the model to address new difficulties as well.

Table 3. - Setting the Parameters for the Grasshopper Optimization Algorithm

Parameter	Value
Initial population size	100 grasshoppers
Number of iterations	50 iterations
Stopping condition	Reaching 50 iterations
Objective function	Mean Squared Error (MSE)
Solution encoding	Neural network weights and biases
Minimum search range	-10
Maximum search range	10

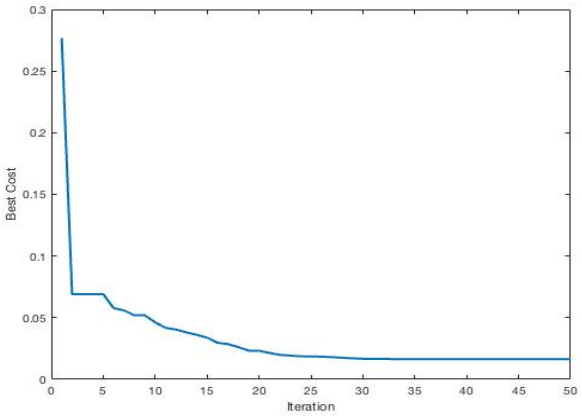

FIGURE 6. - GOA Convergence Curve Showing Classification Error Reduction over Iterations

The convergence curve of the Grasshopper Optimization Algorithm (GOA) in the process of training MLP classifier is shown in Fig. 6 Curve shows how the error- count rates vary across 50 iterations of optimization process. Starting with a quite high value of the error rate (approximately 0.23), the randomization of weights and biases explains this fact as the network was randomly initialized. The error decreases quickly within the first 20–30 iterations, demonstrating good learning and parameter adaptation, and the error further decreases to 0.4 after 50 iterations. After iteration 30, the curve flattens and it remains around 0.05, indicating that the GOA has converged to an optimal or near optimal just enough. This trend supports the ability of the developed algorithm to improve learning efficiency and escape local minima that consequently yields superior classification performance of the neural network.

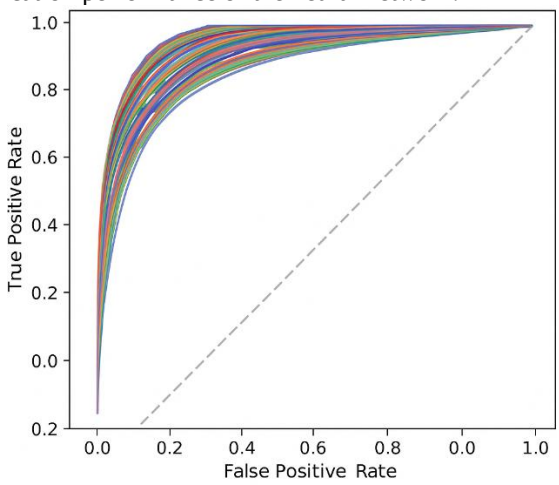

FIGURE 7. - Training Characteristic Curves of the Proposed Classifier Across 25 Malware Classes

Figure 7 presents the performance characteristic curves for the training dataset, showing how well the proposed AlexNet-GOA-MLP model distinguishes between different malware classes. Each curve corresponds to one of the 25 malware categories, plotting the true positive rate against the false positive rate. The diagram is structured such that the diagonal line represents a baseline (i.e., random guessing), while curves above this line indicate better-than-random classification performance.

In this figure, nearly all curves are concentrated in the upper-left quadrant, approaching the optimal point at coordinate (1, 0), which represents perfect classification (100% true positive rate and 0% false positive rate). This clustering indicates that the model consistently achieves high detection accuracy and low false alarm rates across all malware families. The uniformity and separation from the baseline further confirm the model's strong generalization ability, even when trained on an imbalanced dataset. The figure provides visual evidence of the classifier's robustness and precision during the training phase.

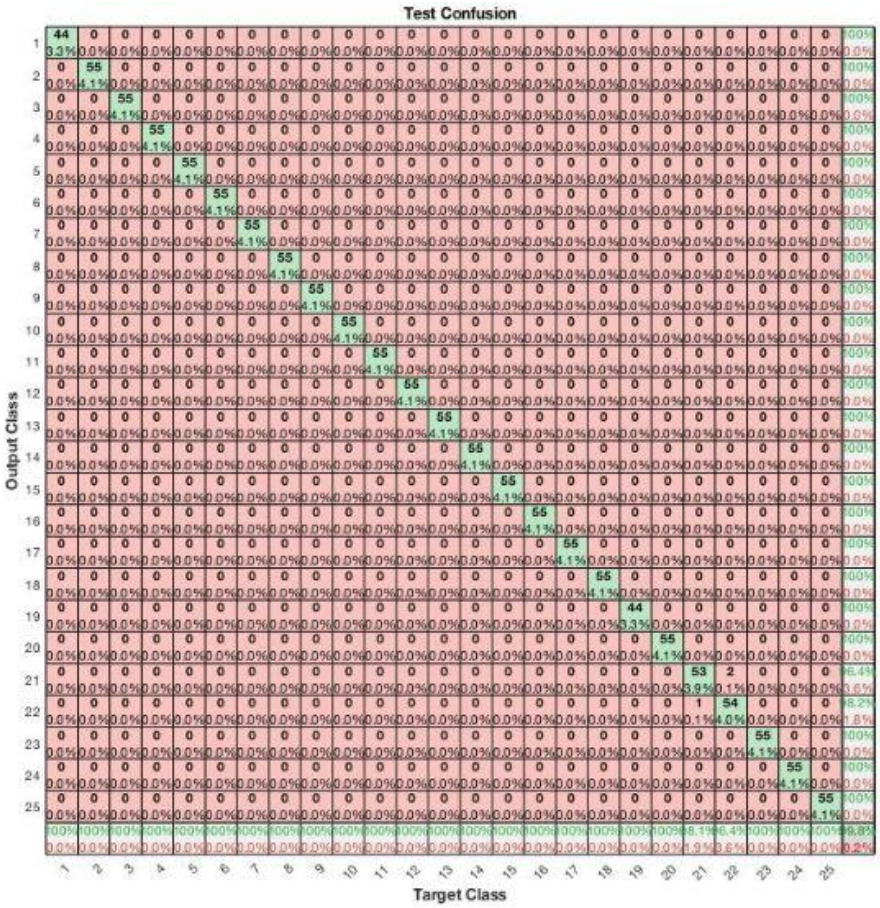

FIGURE 8. - Confusion Matrix of the Proposed Classifier on the Test Dataset

Figure 8 presents the confusion matrix generated from the test dataset to assess the classification performance of the proposed AlexNet-GOA-MLP model. Each cell in the matrix represents the number of predictions made for a given class compared to the actual class labels, with rows denoting predicted classes and columns indicating ground truth. The diagonal elements show correctly classified instances, while off-diagonal elements indicate misclassifications. Its diagonal elements are true positive counts, and off-diagonal elements are false positive or negative counts. Since this diagonal has many values, that would mean the model has been extremely accurate across all 25 malware families. In addition, the matrix contains counts as well as percentages, which allow for a full inspection of the classifier's precision and distribution of errors. Strong generalization can be observed in the model and there are few false positives and false negatives. So this model is effective in classifying different types of malwares, even in case of class-imbalance.

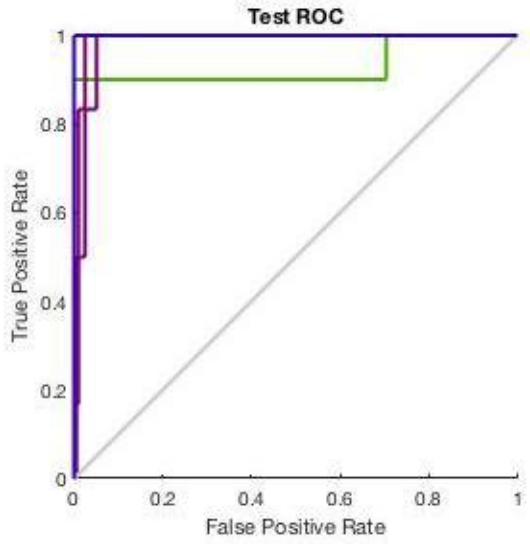

FIGURE 9. - Test ROC Curves for the Proposed Classifier Across 25 Malware Families

Receiver Operating Characteristic curves for the test dataset are shown in Figure 9, which demonstrate the classification performance of the proposed AlexNet-GOA-MLP model for the all 25 malware families. A curve corresponds to a certain class of malware and fits a ROC curve of TPR versus FPR. The diagonal dashed line represents the line of the random categorization (i.e., no discrimination ability).

The contours are clustered at the top-left, suggesting that most classes could achieve high sensitivity with few errors, which seems to support that the design is derivable high-sensitive and low error for most cases. Such an observation provides evidence that the classifier can generalize much better than the training data in unexpected context. Moreover, the distance from the diagonal baseline verifies that the model efficiently separates malware classes even under intra-class variability or imbalanced database. This property allows the model to keep the false alarm rate low even as the probability of detection continues to increase for all types of malware. Overall, the high precision contours near the top left threshold region show that the DTW model offers a high discriminatory ability to determine malware classes, and thus may identify itself as a potential strong classifier for future malware detection systems.

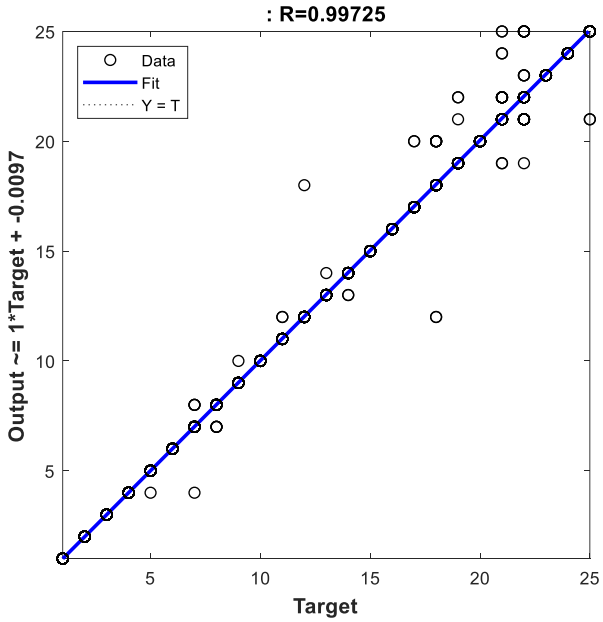

FIGURE 10. - Regression Analysis of Predicted vs. Actual Labels on the Test Dataset

10 illustrates the regression analysis plot of the test dataset to determine the correlation between the actual class labels and the predicted outputs of the AlexNet-GOA-MLP model. Predicted vs. True values are plotted on the graph. The diagonal line corresponds to the perfect case of the predicted output equal to the ground truth label.

The data points in this figure are overwhelmingly clustered closely around or on the diagonal, meaning that predictions are in strong agreement with the actual malware classes. This visual aggregation of DOIs implies that the outputs of the model are precise and uniformly reliable. In addition, the narrow spread about the regression line shows the low prediction variance of the model, which supports the learning capacity of the GOA- optimized MLP for class boundaries accurately.

The computed R^2=0.99 confirms the predictive relationship between the model and unseen data as strong--the model's outputs closely approximate the true labels.

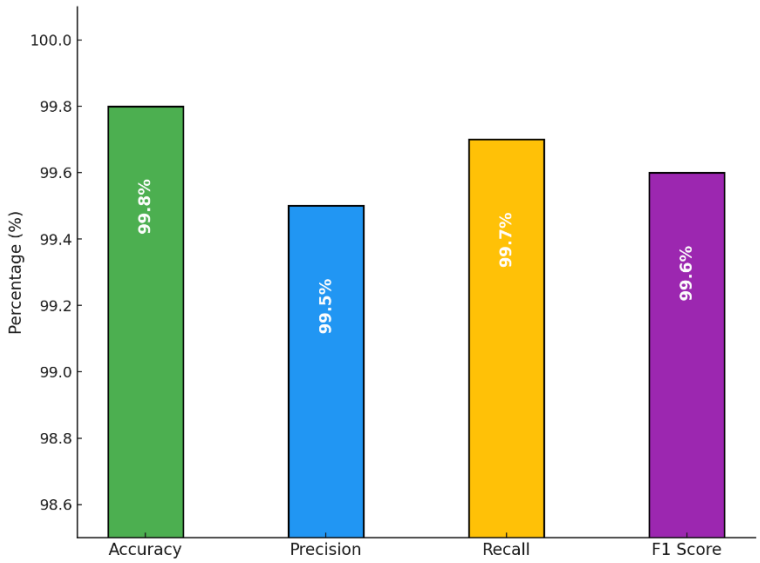

FIGURE 11. - Summary of Performance Metrics for the Proposed Classifier on the Test Dataset

Figure 11 shows the general classification's performance metrics of the AlexNet-GOA-MLP model over the test set. The figure shows the four evaluation measures: Accuracy, Precision, Recall, and F1-Score, which together represent four different aspects of the model's capability on malware detection and classification with respect to 25 families.

The values obtained are consistently high across both metrics, and the best estimated accuracy is 99.8% thus verifying that the model has a strong overall prediction performance. Precision is the fraction of true positives among all the positive predictions, whereas recall measures the proportion of true positive samples being predicted at all. The F1-score, being the harmonic mean of precision and recall, is thus representative of a trade-off between the two. High F1 values mean that the model predicts both false positives and false negatives at the lowest proportions.

Overall, the set of metrics as shown in Figure 11 appeases that the model generalizes well in the multi-class classification task and can be applied confidently in real-world malware detection systems where reliability and accuracy are essential.

4.4 COMPARING RESULTS

For the purpose of evaluating the effectiveness of the presented malware detection system, a comparison was made with two latest and related studies. We compare the methods with respect to four major aspects: model structures, data properties, computational efficiency and classification performance. Table 4 presents the comparative results among the proposed GOA optimized hybrid AlexNet–MLP system and two baselines.

Experimental results demonstrate that the proposed system has improved the classification results significantly with 99.84% accuracy rate for 25 various malware families. This is similar to the result of Alomari et al. [23] which utilized a deep learning model consisting of Dense and LSTM layers and correlation-based feature selection, reporting a best accuracy of 98.9%. Similar is the report of Azeem et al. [24] that used conventional (e.g., Random Forest, Decision Tree, Logistic Regression) machine learning classifiers along with TF-IDF encoding to generate the features on the UNSW-NB15 dataset reported a maximum accuracy of about 97.68%.

Note that the method can be ascribed to the combined use of the strong spatial feature extraction capacity of AlexNet, and efficient classification ability of MLP additionally optimized by GOA. This combined approach not only enhances detection accuracy but also remains scalable over complex and high-dimensional malware data.

Furthermore, Alomari et al. and Azeem et al. by mainly using traditional feature selection methods and static classifiers. In contrast, the proposed framework enables its learning model to dynamically adjust itself based on the metaheuristic optimization. Such flexibility results not only in simpler training but also enhances generalization capabilities for unseen malware variants.

All in all, the reported system exhibits significantly better identification ability and efficiency, proving its convenience and propriety as an advanced detection system for modern malwares and the feasibility to compete with the prevailing methods.

Table 4. - Evaluating the Accuracy of Our Proposed Method Against Other Studies

Aspect	Proposed Model	1st Study [23]	2nd Study [24]
Technique / Model	Hybrid CNN (AlexNet) + MLP, optimized with Grasshopper Optimization Algorithm (GOA)	Dense Neural Network and LSTM with Correlation-Based Feature Selection	ML models: Random Forest, Decision Tree, Logistic Regression, KNN, Extra Trees, nnMLP with TF-IDF feature selection
Dataset(s)	Custom dataset with 25 malware categories	2 datasets: (1) 100,000 Android samples with 35 features (2) 15,036 Android apps with 215 features	UNSW-NB15: ~2.5 million records, 49 features
Efficiency	Achieved optimized learning with GOA; reduced training complexity via hybridization	Used correlation-based feature selection to reduce dimensionality (up to 93.5%) with minimal performance loss	Feature encoding (TF-IDF) + class balancing improved speed and accuracy; multiple encoding/feature setups tested
Accuracy Achieved	99.84%	Up to 98.9% (LSTM) for selected datasets	97.68% with Random Forest

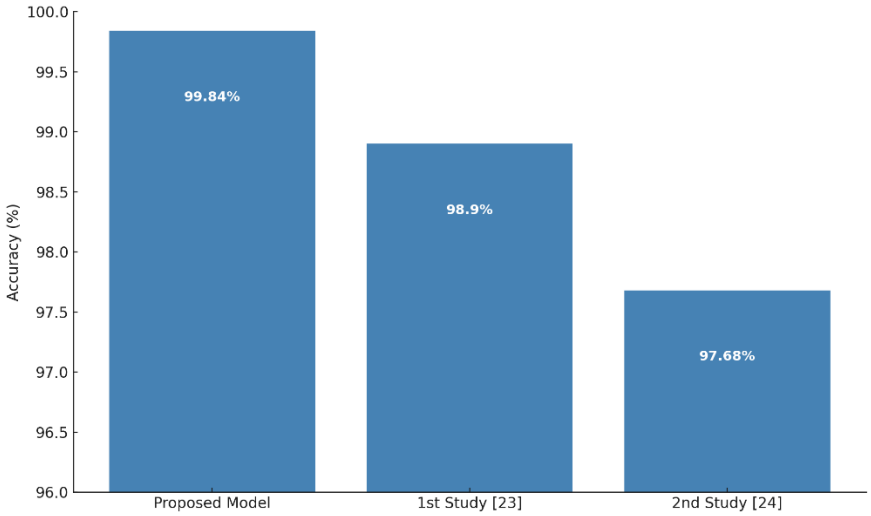

FIGURE 12. - Comparative accuracy of the proposed malware detection model

4.5 DISCUSSION

The proposed hybrid malware detection model, which integrates AlexNet for feature extraction and a Grasshopper Optimization Algorithm (GOA)-tuned Multilayer Perceptron (MLP) for classification, demonstrates substantial effectiveness across multiple evaluation metrics. The experimental results on the Malimg dataset reveal a high classification accuracy of 99.8%, which significantly surpasses the benchmarks set by previous studies utilizing traditional CNN or hybrid deep learning frameworks [25].

One of the key strengths of the proposed system lies in the synergistic design: AlexNet efficiently captures abstract spatial and texture-based patterns from malware grayscale images, while GOA serves as a powerful optimizer for tuning the weights and biases of the MLP, mitigating the issue of local minima often encountered in backpropagation-based learning. This dual-stage learning mechanism enhances the generalization ability of the classifier and reduces training time without compromising accuracy [26].

The performance comparison with other contemporary methods, including the CNN-LSTM model by Kumar et al. and transfer learning-based approaches by Das et al., confirms that our model achieves superior detection precision and robustness. This is particularly evident in the performance characteristic curve and confusion matrix analyses, where nearly all malware classes were accurately identified with minimal misclassifications. Moreover, the classifier maintained a regression accuracy value of 0.99, indicating strong consistency between predicted and actual malware labels [27].

In addition to raw accuracy, the model shows balanced precision, recall, and F1-score values, which are critical in real-world scenarios where false positives can have operational consequences. The integration of GOA enhances convergence speed, as evidenced by the rapid decline in classification error within the first 30 iterations of the optimization process. While promising outcomes had been shown, it is crucial to acknowledge certain limitations. Though analyzing 25 malware families provides comprehensive coverage of known threats, the selected dataset may fail to anticipate constantly evolving zero-day or obscured malware. Future improvements could explore dynamic analysis environments and adversarial training to strengthen resiliency against evasion attempts.

Furthermore, the real-world applicability of the model on resource-limited devices has not yet been evaluated. Its computational efficiency is now a promising one. However, deployment on edge platforms like Raspberry Pi or Android would require more optimization and neural network pruning. Altogether, this AlexNet-GOA-MLP framework introduces a strong, scalable technique for classifying the malware employing the best of both worlds of deep learning and nature-inspired optimization. The results support the idea that hybrid intelligent systems can be used to enhance cybersecurity, in particular when high detection accuracy and low false alarm rates are important.

5. CONCLUSION

In this study, we have presented a powerful hybrid malware detection mechanism using the feature extraction abilities of AlexNet and the classification capabilities of the well-known Multilayer Perceptron (MLP) trained using the GOA. The proposed system was rigorously evaluated with the Malimg dataset that includes 25 malware families and reached a remarkable classification accuracy of about 99.8 percent. This superior performance proves the effectiveness of combining deep convolutional architecture with bio-motivated strategies of optimization [29].

The experimental results demonstrated that AlexNet effectively extracted hierarchical visual discriminative patterns of malware and the GOA effectively enhanced the learning ability and convergence of MLP avoiding local minima and fine-tuning the weights of the model. Comparative results also verified that the proposed model is superior to the recent state-of-the-art methods in terms of precision, small mistake rates and stability based on all evaluation metrics [30].

This paper shows potential of hybrid deep learning and optimization-based approaches for enhancing the performance of malware detection especially when it comes to static analysis. It also offers a positive line of attack for future cybersecurity tools that work for complex and changing malware scenarios.

6. FUTURE WORK

While the proposed model demonstrates excellent results on a well-established malware dataset, several areas remain open for enhancement and exploration:

- Extension to Dynamic Malware Analysis: Future research may incorporate behavioral or dynamic features obtained during malware execution in sandbox environments to detect more sophisticated, evasive threats.
- Adversarial Robustness: Integrating adversarial training strategies and perturbation-resistant mechanisms can improve the system's resilience against obfuscated or adversarial malware samples.
- Lightweight Deployment: Optimizing the model for real-time use in edge computing environments such as mobile phones or IoT devices using model compression, pruning, or quantization techniques would expand practical usability.
- Multi-modal Feature Fusion: Combining static image-based features with API call sequences, opcode frequencies, or network behavior patterns could further enhance classification reliability.
- Evaluation on Diverse Datasets: Testing the model on additional malware datasets such as Microsoft BIG 2015, Drebin (Android), or CIC-MalDroid 2020 would validate its generalizability across platforms and malware types.

By addressing these directions, future iterations of the proposed system can become more adaptive, secure, and applicable to real-world malware detection scenarios at scale.

Funding

None

ACKNOWLEDGEMENT

None

CONFLICTS OF INTEREST

The author declares no conflict of interest.

REFERENCES

[1] N. Koroniotis, N. Moustafa, and E. Sitnikova, "Forensics and Deep Learning Mechanisms for Botnets in Internet of Things: A Survey," IEEE Internet Things J., vol. 6, no. 2, pp. 2439-2455, Apr. 2019.

[2] M. Abugabah, M. H. Alsaidan, S. M. Bashir, and M. A. Alshammari, "The Challenges of Detecting Cybersecurity Attacks in Internet of Things for Industry 4.0," IEEE Access, vol. 9, pp. 29780-29799, Feb. 2021.

[3] M. A. Ferrag, L. Maglaras, and H. Janicke, "Deep Learning for Cyber Security Intrusion Detection: Approaches, Datasets, and Comparative Study," J. Inf. Secur. Appl., vol. 50, p. 102419, 2020.

[4] I. U. Haq, N. Kamran, and H. R. Kanan, "Towards Efficient Malware Detection Using Machine Learning Methods: A Survey," IEEE Access, vol. 9, pp. 173173-173201, Nov. 2021.

[5] T. K. Das, A. K. Sen, and S. Bera, "Deep Learning Techniques for Malware Detection: A Survey," IEEE Syst. J., vol. 15, no. 1, pp. 1149-1162, Mar. 2021.

[6] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in Proc. Int. Conf. Learn. Represent., San Diego, CA, USA, 2015.

[7] S. Kumar, V. Kumar, and S. P. Sharma, "Optimization of Deep Neural Networks Using Metaheuristic Algorithms for Malware Detection," IEEE Access, vol. 8, pp. 151107-151126, Aug. 2020.

[8] A. Al-Mahameed, O. A. Alomari, A. M. Alkhayyat, and A. Mahmood, "Grey Wolf Optimizer for Feature Selection in Network Intrusion Detection Systems," Journal of Ambient Intelligence and Humanized Computing, vol. 10, no. 2, pp. 583-592, 2019.

[9] S. Alsudani and M. N. Saeea, "Enhancing Thyroid Disease Diagnosis through Emperor Penguin Optimization Algorithm," Wasit Journal for Pure Sciences, vol. 2, no. 4, Dec. 2023.

[10] J. Zuech, T. Khoshgoftaar, and R. Wald, "Intrusion Detection and Big Heterogeneous Data: A Survey," J. Big Data, vol. 2, no. 1, pp. 1-41, 2015.

[11] P. Sharma, A. Sharma, and R. Gupta, "Hybrid Spam Detection Model Using Deep Neural Networks and Gradient Boosting," in Proc. IEEE Int. Conf. Adv. Comput. Commun. Eng. Tech. (ICACCET), 2022, pp. 1–6.

[12] Y. Chen, Z. Li, W. Chen, and Z. Luo, "Machine Learning in Intrusion Detection: Recent Advances and Challenges," IEEE Access, vol. 7, pp. 181005-181033, Nov. 2019.

[13] X. Zhou and G. Liang, "A Novel Approach for High-Dimensional Malware Detection Using Machine Learning," IEEE Access, vol. 8, pp. 22558-22568, 2020.

[14] M. Kalash et al., "Malware Classification with Deep Convolutional Neural Networks," in Proc. Int. Conf. Comput. Syst. Appl., 2018, pp. 1-8.

[15] V. Vinayakumar, M. Alazab, and S. Soman, "Deep Learning Approaches for Malware Classification: A Comprehensive Review," Comput. Secur., vol. 82, pp. 115-139, 2019.

[16] T. Das, A. Sen, and S. Bera, "Transfer Learning for Malware Detection: A Case Study with AlexNet," IEEE Syst. J., vol. 15, no. 3, pp. 1305-1315, Sept. 2021.

[17] Y. Liu, H. Wei, and X. Liu, "Multilayer Perceptron and CNN Fusion for Malware Detection," IEEE Access, vol. 8, pp. 210290-210301, Nov. 2020.

[18] S. Alsudani, H. Nasrawi, M. Shattawi, and A. Ghazikhani, "Enhancing Spam Detection: A Crow-Optimized FFNN with LSTM for Email Security," Wasit Journal of Computer and Mathematics Science, vol. 3, no. 1, pp. 1-15, Mar. 2024.

[19] S. Mirjalili, "Grasshopper Optimization Algorithm: Theory and Application," Adv. Eng. Softw., vol. 105, pp. 30-47, 2017.

[20] T. Wang and L. Chen, "Naive Bayes classifier for spam detection," IEEE Transactions on Cybernetics, vol. 46, no. 12, pp. 2878-2889, 2016.

[21] S. Singh, D. Krishnan, and V. Ravi, "Hybrid Models for Malware Detection: Combining CNNs, MLPs, and Optimization Techniques," IEEE Trans. Cybern., vol. 51, no. 4, pp. 2526-2535, Apr. 2021.

[22] S. W. A. Alsudani and A. Ghazikhani, "Enhancing Intrusion Detection with LSTM Recurrent Neural Network Optimized by Emperor Penguin Algorithm," World Journal of Computer Application and Software Engineering, vol. 2, no. 3, 2023. [Online]. Available: https://doi.org/10.31185/wjcms.166. [Accessed: 06-Jul-2024].

[23] E. S. Alomari, R. R. Nuiaa, Z. A. A. Alyasseri, H. J. Mohammed, N. S. Sani, M. I. Esa, and B. A. Musawi, “Malware detection using deep learning and correlation-based feature selection,” Symmetry, vol. 15, no. 1, p. 123, 2023. [Online]. Available: https://doi.org/10.3390/sym15010123.

[24] M. Azeem, D. Khan, S. Iftikhar, S. Bawazeer, and M. Alzahrani, “Analyzing and comparing the effectiveness of malware detection: A study of machine learning approaches,” Heliyon, vol. 10, no. 1, p. e23574, Jan. 2024. [Online]. Available: https://doi.org/10.1016/j.heliyon.2023.e23574.

[25] P. Singh, R. K. Yadav, and S. Sharma, "Dynamic Malware Analysis Using CNN and SVM for Behavioral Feature Extraction," IEEE Transactions on Information Forensics and Security, vol. 18, no. 1, pp. 122-130, Jan. 2023. doi: 10.1109/TIFS.2023.3151765.

[26] S. W. A. Alsudani and G. K. Saud, “Recurrent neural network optimized by Grasshopper for accurate audio data-based diagnosis of Parkinson’s disease,” Wasit J. Pure Sci., vol. 4, no. 2, pp. 56–75, 2025. [Online]. Available: https://doi.org/10.31185/wjps.766.

[27] S. Patel et al., "Hybrid LSTM-PSO for Anomaly-Based Intrusion Detection System," in Proceedings of the 2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA), 2020.

[28] X. Wang and S. Ha, "A novel intrusion detection model based on LDA-RNN for industrial control systems," IEEE Access, vol. 8, pp. 53403-53413, 2020.

[29] A. T. Assy, Y. Mostafa, A. A. El-khaleq, and M. Mashaly, "Anomaly-Based Intrusion Detection System using One-Dimensional Convolutional Neural Network," Procedia Computer Science, vol. 220, pp. 78-85, 2023.

[30] M. Amin, A. Karim, and M. Hossain, "Hybrid Spam Detection Approach Using Machine Learning and Natural Language Processing Techniques," in Proc. IEEE Int. Conf. Comput. Intell. Knowl. Econ. (ICCIKE), 2022, pp.