Library Progress International *Print version ISSN 0970 1052*
Vol.44 No. 3, Jul-Dec 2024: P. 16054-16059 *Online version ISSN 2320 317X*

Original Article *Available online at www.bpasjournals.com*

Malware Detection Using Convolutional Neural Network and Perceptron Neural Network Optimized with Firefly Algorithm

Fatimah abd-alrodh rashed[1], Rehab k. Kadhim[2], Ali Adnan AL-KHAZRAJI[3], Nour Sadiq Abdulqadir[4], Malik A. Alsaedi[5]

[1] Department of Computer Engineering, Collage of Engineering, Al-Iraqia University, Baghdad- Iraq.
[2] Department of Computer Engineering, Collage of Engineering, Al-Iraqia University, Baghdad- Iraq.
[3] Department of Computer Engineering, Collage of Engineering, Al-Iraqia University, Baghdad- Iraq.
[4] Department of IT,Administrative information business management ,Al-Iraqia University , Baghdad- Iraq.
[5] Department of Electrical Engineering , Collage of Engineering ,Al-Iraqia University , Baghdad- Iraq
fatimah.a.rashid@aliraqia.edu.iq, rehab.k.kadhim@aliraqia.edu.iq, Ali.adnan@aliraqia.edu.iq , noor.s.abdulqadir@aliraqia.edu.iq, maliksaady@yahoo.com

How to cite this article: Fatimah Abd-Alrodh Rashed, Rehab k. Kadhim, Ali Adnan AL-Khazraji, Nour Sadiq Abdulqadir, Malik A. Alsaedi (2024) Malware Detection Using Convolutional Neural Network and Perceptron Neural Network Optimized with Firefly Algorithm. *Library Progress International,* 44(3), 16054-16059

***Abstract*—**In this paper, in order to detect 25 classes of malware, with the aim of increasing the detection accuracy, we used the pre-trained convolutional neural network of Alex Net and combined it with the perceptron neural network optimized with the Worm Shabbat algorithm. In fact, Alex Net's convolutional neural network automatically extracted 1000 feature vectors for each input image using the convolutional layer in its architecture. In the next step, we used the transfer learning method to classify the extracted features. In this thesis, we transferred the learning done by the Alex net convolutional neural network to a multi-layer perceptron neural network that was optimized using the firefly meta-heuristic algorithm for classification. In this work, we optimized the optimal weight and bias of the neural network by meta-heuristic algorithm. Finally, we were able to achieve 99.8% accuracy, which showed that the proposed method was superior in terms of accuracy compared to the compared methods.
***Index Terms*—**convolutional neural network, firefly algorithm, meta-heuristic algorithm, line of sight, etc.

1. INTRODUCTION

Malware means software whose purpose is to damage and infiltrate the computer, and this is done in a situation where the owner of the system has neither knowledge nor consent. Today, malware is considered as an important threat to the security and integrity of information. Malware is becoming more and more sophisticated all the time. After entering a system, malware can perform actions such as sending spam emails, stealing information and passwords, etc. Different methods have been considered to detect malware, one of the best methods is using machine learning [1].
Security flaws caused by malicious software attacks have increased security concerns in the digital age. Problem Statement Since most computer users, companies, and governments are affected by the dramatic growth of malware attacks, malware detection is an important research topic. In this thesis, in order to detect 25 classes of malware as mentioned in table 1, with the aim of increasing the detection accuracy, he used the pre-trained convolutional neural network of Alex Net and combined it with the perceptron neural network optimized with the firefly algorithm. Examining data mining solutions in identifying malware [2-3].

- **Providing new methods to detect malware based on data mining algorithms.**
- **Finding a solution that can process programs and extract its features and predict whether the program under the process is healthy programs or malware. Main goals Micro goals.**
- **Using data mining methods to detect the penetration of malware into computer devices**

2. MEHTODOLOGY

Get image preprocessing, resizing the image using the pre-trained neural network of alexnet in order to extract features from images not using fully connected layers and removing fully connected layers and transferring learning to perceptron neural network the initial value for the weight and bias of the neural network based on perceptron training and optimization stage evaluate values using the objective function update weight and bias value using reaching the maximum iteration of the algorithm evaluation of trained neural network [4].
One of the advantages of using convolutional neural networks to extract optimal features is that this neural network is not sensitive to darkness or brightness or noise due to the use of convolution operations, and it is only enough that the

dimensions of the input images are proportional to the input of the neural network i.e. 224* be 224 For this reason, in this thesis, for the pre-processing stage, we will only change the dimensions of the images to 224x224 and prepare the images for the feature extraction stage.

Step of feature extraction from images, to extract optimal features in this work, convolutional neural network with Alex net architecture has been used. Alex Net is a deep convolutional neural network that is presented for recognition and classification of color images with size 224x224x3. This neural network has 62 million learning parameters and 11 layers [5]. This network is one of the pre-trained convolutional networks. Pre-trained means that this network has already been trained on the ImageNet dataset and on thousands of different images, and the parameters of this neural network have been set, and to use it, it is enough to give the images as input to this network so that the features Extract images. These networks have solved the common problems in conventional convolutional neural networks that need to be trained by millions of images.

Preparing extractive features for transfer learning At this stage, before we transfer the data to the optimized perceptron for classification, the data will be normalized and then divided into two groups of testing and training with a ratio of 70 to 30. In general, normalization makes the impact of all features in the neural network training process to be the same, and no data is preferred over data with a lower value because of a higher value. After the extracted features are normalized, 70% of the data is used to train the classifier and 30% is used to evaluate the performance of the classifier.

Training and optimization of multi-layer perceptron using wormhole algorithm [6]. In this step, we use multilayer perceptron neural network to classify the features, and the optimal value of weight and bias of this neural network is set using the firefly optimization algorithm. Perceptron neural network consists of three main layers namely input layer, hidden layer and output layer. Each layer has neurons depending on the dimensions of the data set. In most researches, the number of neurons in the hidden layer, which is the main processing layer, is determined by trial and error. In this neural network, each layer has an activation function, and the most famous activation function used in perceptron neural network is the sigmoid activation function. This activator function receives a number with a real value as input and takes it to the interval between zero and one.

Training and optimization of multi-layer perceptron using wormhole algorithm. In this step, we use multilayer perceptron neural network to classify the features, and the optimal value of weight and bias of this neural network is set using the firefly optimization algorithm. Perceptron neural network consists of three main layers namely input layer, hidden layer and output layer. Each layer has neurons depending on the dimensions of the data set. In most researches, the number of neurons in the hidden layer, which is the main processing layer, is determined by trial and error. In this neural network, each layer has an activation function, and the most famous activation function used in perceptron neural network is the sigmoid activation function. This activator function receives a number with a real value as input and takes it to the interval between zero and one.

Training and optimization of multi-layer perceptron using wormhole algorithm Perceptron neural network uses error back propagation algorithm in non-optimal mode for training. In the error back propagation algorithm in the perceptron network, the signal moving in the forward direction changes the weights and bias coefficients, and after the network output is obtained, this output is compared with the real value and an error is obtained. In the second step, the signal that moves on the return path, which is known as the error signal, changes the weight and bias of the layers according to the error value, and the training process starts again [7]. This process continues until the algorithm reaches a low error and stops.

Training and optimization of multi-layer perceptron using wormhole algorithm one of the problems of the error back propagation method for neural network training is getting stuck in local optima. This means that sometimes the amount of error obtained in the process of training the network may be small and the algorithm stops, if one or more iterations continue, it achieves a smaller amount of error and the accuracy of the network increases. To solve this problem in this thesis, instead of using the error back propagation algorithm, the firefly optimizer algorithm is used to train and adjust the weight and bias of the neural network [8].

The experimental data used in this thesis is from the large-scale and unbalanced Windows malware dataset (Malimg). This dataset contains 9339 malware image samples from 25 families with 80 to 2949 samples per family, the families belong to the following categories: Worm, PWS, Dialer, Rogue, Backdoor, Trojan and TDownloader.

Table 1: Description of the Malimg dataset families

S. No	Class	Family	Nos
1.	Worm	Allaple. L	1591
2.	Worm	Allaple. L	2949
3.	Worm	Yuner. A	800
4.	PWS	Lolyda AA1	213
5.	PWS	Lolyda AA2	184
6.	PWS	Lolyda AA3	123
7.	Trojan	C2Lop.P	146
8.	Trojan	C2Lop.gen!g	200
9.	Dialer	Instantaccess	431

10.	TDownloader	Swizzot.gen!I	132
11.	TDownloader	Swizzot.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	Sialplatform B	177
20.	TDownloader	Dontovo A	162
21.	TDownloader	Obfuscator.AD	142
22.	Backdoor	Agent.FYI	116
23.	Worm:AutoIT	Autorun K	106
24.	Backdoor	Rbot!gen	158
25.	Trojan	Skintrim N	80

3. EVALUATION CRITERIA

The criteria used in this work to evaluate the proposed model are: accuracy, recall, accuracy. Each of the mentioned criteria evaluates the classifier from different aspects

$$Acuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

$$precision = \frac{TP}{TP+FP} \quad (3)$$

4. ANALYSIS AND REVIEW OF THE PROPOSED METHOD

According to the opposite figure, it can be seen that in the neural network, the number of neurons in the input layer is equal to the number of features extracted for each sample, i.e. 1000, and the number of neurons in the output layer is equal to 25 neurons per the number of output classes [9]. Finally, the number of hidden layer neurons is equal to 16 neurons using trial and error.

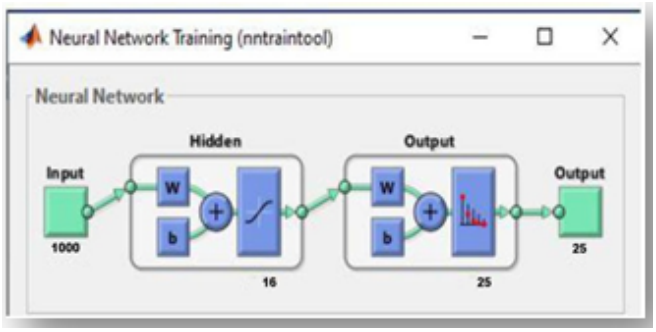

Figure 1: Neural network architecture used for data classification

The activator functions used for the sigmoid hidden layer have been determined and the training of this network has been done based on the firefly algorithm whose specifications are given in the opposite table [10].

5. RESULTS AND DISCUSSION

As can be seen, the error rate was very high at the beginning of the work, and the error rate decreased gradually with the updating of the population towards members with greater fitness. From the 30th iteration onwards, the error rate has almost reached its lowest value in the neural network training process.

According to the figure, it can be seen that the error rate for the optimized perceptron classifier has reached less than 0.05, which indicates that the perceptron neural network is well trained.

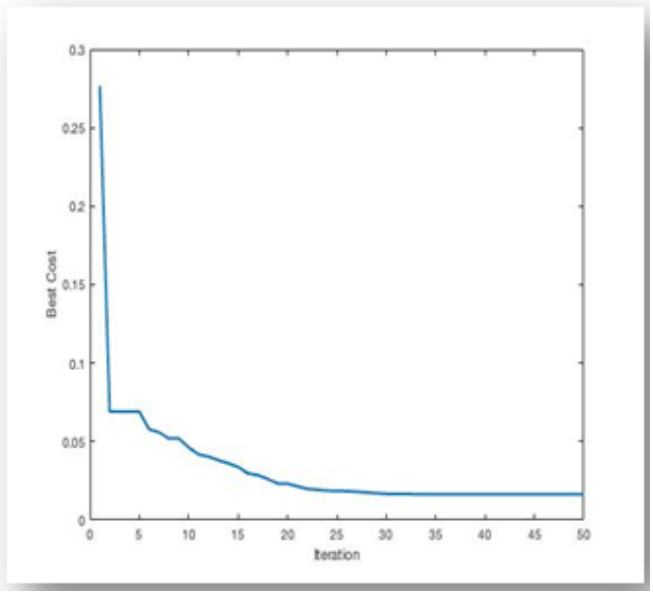

Figure 2: Convergence diagram of wormhole algorithm during neural network optimization process

As can be seen, the error rate was very high at the beginning of the work, and the error rate decreased gradually with the updating of the population towards members with greater fitness. From the 30th iteration onwards, the error rate has almost reached its lowest value in the neural network training process.
According to the figure, it can be seen that the error rate for the optimized perceptron classifier has reached less than 0.05, which indicates that the perceptron neural network is well trained.
As can be seen, all the data are almost in line with the blue line and have only a small amount of scatter. Also, regression has a value between zero and one, the closer the match between the neural network output and the real output is, the closer the regression value will be to one, and on the contrary, the lower the match between the neural network output and the real output, the closer the regression value will be to zero. According to the figure, it can be seen that the regression value is equal to 0.99.

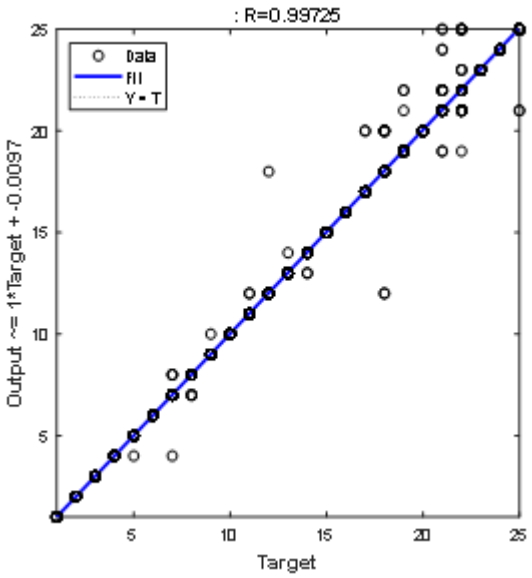

Figure 3: Regression plot for test data

Analysis and review of the results of the proposed method
In the opposite figure, the results obtained for the test data set are examined from different aspects in the form of a bar graph.

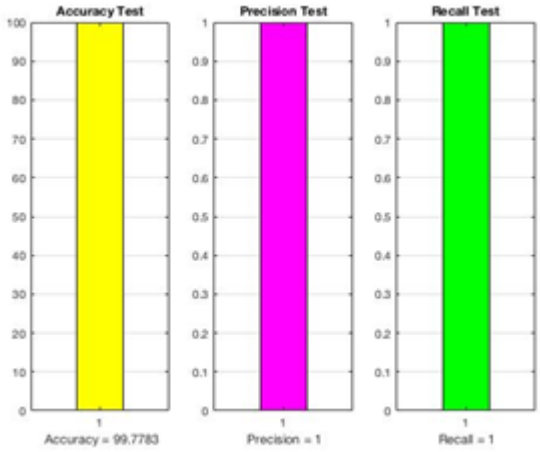

Figure 4: Evaluation criteria for test data

On average, the proposed method is 1% more accurate than the other methods in the table. In general, according to the obtained results, it can be concluded that in this thesis, we were able to extract the optimal features using the convolutional neural network of Alexnet and determine the weight and bias of the perceptron neural network well in the classification stage, which ultimately leads to an increased Accuracy.

Table 2: Performance Metrics

S. No	Article	Method	Accuracy criterion
1.	Reference [1]	Normal convolutional neural network	98%
2.	Reference Article[2]	Random Forest	98.5%
3.	Proposed Method	Alex Net + Perceptron + Worm Shabtab	**99.8%**

In this article, in order to detect 25 classes of malware, with the aim of increasing the detection accuracy, we used the pre-trained convolutional neural network of AlexNet and combined it with the perceptron neural network optimized with the Worm Shabtab algorithm
In this article, we transferred the learning done by the Alexnet convolutional neural network to a multi-layer perceptron neural network that was optimized using the firefly meta-heuristic algorithm for reflections.

6. Conclusion

In this paper, in order to detect 25 classes of malware, with the aim of increasing the detection accuracy, we used the pre-trained convolutional neural network of AlexNet and combined it with the perceptron neural network optimized with the Worm Shabtab algorithm.
In this paper, we transferred the learning done by the alexnet convolutional neural network to a multi-layer perceptron neural network that was optimized using the firefly meta-heuristic algorithm for classification.
In this work, we optimized the optimal weight and bias of the neural network by meta-heuristic algorithm. Finally, we were able to achieve 99.8% accuracy, which compared to the compared methods, the proposed method was superior in terms of accuracy.

References

[1] Idika, N., & Mathur, A. P. (2007). A survey of malware detection techniques. Purdue University, 48(2), 32-46.
[2] Christodorescu, M., & Jha, S. (2004). Testing malware detectors. ACM SIGSOFT Software Engineering Notes, 29(4), 34-44.
[3] McLaughlin, N., Martinez del Rincon, J., Kang, B., Yerima, S., Miller, P., Sezer, S., ... & Joon Ahn, G. (2017, March). Deep android malware detection. In Proceedings of the seventh ACM on conference on data and application security and privacy (pp. 301-308).
[4] Saeed, I. A., Selamat, A., & Abuagoub, A. M. (2013). A survey on malware and malware detection systems. International Journal of Computer Applications, 67(16).
[5] Ye, Y., Li, T., Adjeroh, D., & Iyengar, S. S. (2017). A survey on malware detection using data mining techniques. ACM Computing Surveys (CSUR), 50(3), 1-40.
[6] Sahs, J., & Khan, L. (2012, August). A machine learning approach to android malware detection. In 2012 European Intelligence and Security Informatics Conference (pp. 141-147). IEEE.

[7] Roundy, K. A., & Miller, B. P. (2010, September). Hybrid analysis and control of malware. In International Workshop on Recent Advances in Intrusion Detection (pp. 317-338). Springer, Berlin, Heidelberg.
[8] Morales, J. A., Al-Bataineh, A., Xu, S., & Sandhu, R. (2010, September). Analyzing and exploiting network behaviors of malware. In International conference on security and privacy in communication systems (pp. 20-34). Springer, Berlin, Heidelberg.
[9] S. H. Alnajjar and H. M. Mahmoud, "Internet of Things Utilizing Light Fidelity Technology: A Review ", IJSER, vol. 2, no. 4, pp. 1–8, Dec. 2023.
[10] M. A. Hailan, B. M. Albaker, and M. S. Alwan, "Two-Dimensional Transformation of a Conventional Manufacturer into a Smart Manufacturer: Architectonic Design, Maintenance Strategies and Applications", IJSER, vol. 1, no. 1, pp. 77–87, Sep. 2022.