Published by
NIGERIAN SOCIETY OF PHYSICAL SCIENCES
Available online @ https://journal.nsps.org.ng/index.php/jnsps

J. Nig. Soc. Phys. Sci. 6 (2024) 2192

Journal of the Nigerian Society of Physical Sciences

Development of a machine learning based fileless malware filter system for cyber-security

Umaru C. Obini, Chukwu Jeremiah*, Sylvester A. Igwe

Department of Computer Science, Ebonyi State University, P.M.B. 053, Abakaliki, Nigeria

Abstract

Over the years, the increased rate of perturbated malware based cyber-attack has presented many challenges and triggered the need for immediate solution all over the world. This was addressed in this paper development of a machine learning based fileless malware filter system for cyber security. Fileless malware which can come inform of Memory Resident Fileless malware or Windows Registry fileless malware are known to have no executable files, resides in the system memory or the windows registry. It doesn't write any files to disk making it very challenging to detect using traditional signature-based methods. It often leverage legitimate tools such as Windows Management Instrumentation and PowerShell to carry out its malicious activities. The methods used were data collection, data extraction, Deep Neural Network (DNN), activation function, training algorithm and classification. The methods were designed using structural and mathematical approaches which employed architectural diagrams, flow charts and self-defining equations to develop the new system. The training of the DNN was done using Gradient Descent Algorithms (GDA) to generate the malware filter algorithm. The filters were implemented with Simulink, tested and validated. The results were also evaluated using Regression (R) and Mean Square Error (MSE analyzer) and it showed R values of 0.9931 and MSE performance also recorded 0.002088Mu. This implied that the filter developed was able to detect and remove malware on the network.

DOI:10.46481/jnsps.2024.2192

Keywords: Perturbated, Malware, Cyber-attack, Filter

Article History :
Received: 14 June 2024
Received in revised form: 06 September 2024
Accepted for publication: 10 September 2024
Published: 29 September 2024

Communicated by: Oluwatobi Akande

1. Introduction

Every day begins a story, and today the issue of "cybersecurity" has dominated the topic for discussion in the global research community. Cybersecurity is the application of science and technology for the protection of networks, programs and data from cyber-attacks [1]. This cybersecurity issues have attracted global concern, as even the simplest gadgets available today such as phones, smart watches, laptops, palm tops, cameras, etc. are all dependent on the platform of the internet for operation; not mentioning our institutions, government organizations, private companies among others [2].

Coupled with the robust nature of cloud virtual server, almost all classified information such as company data, financial data, government data, tertiary institution data, military data among others, are stored on network servers for better management [3]. This presents opportunities for terrorists, fraudsters, hackers, etc. to carefully study the fundamental flaws of this network servers, identifying their loopholes and security vul-

*Corresponding author Tel. No.: +234-803-755-0845.
Email address: chukwu.jeremiah@ebsu.edu.ng (Chukwu Jeremiah)

nerabilities, then developing strategies for cyber-attack to steal information and commit other unlawful activities [4].

Prior to the last decade, the possibility of actualizing this unlawful act was difficult and the rate of successful penetration of cyber-attacks were relatively low due to the static nature of the threat model as at then; However, since the year 2011 till date, the rate of malicious attacks has steadily increased and requires urgent attention to be tackled [1]. According to Heenaa *et al.* [5], the success rate of cyber-attacks in the past decade was due to the advancement in the attack model. These hackers employed sophisticated technologies and tools which manipulate (blinds) the conventional security solutions to perpetrate the cyber-attack and cause harm to the network [5].

According to Heenaa *et al.* [5] and Graeber [6], malware is one of the most used cyber-attacks worldwide today. Malware is a malicious software which is used with the intention of breaching computer systems security policies with respect to confidentiality, integrity and availability of data [7]. The choice of malware attack over other attack techniques was due to its adaptive ability and difficulty in detection [8].

Today, countless solutions have been proposed to filter malwares in wireless network. These solutions were classified into two categories which are signature based and behavioral based.

The signature-based solution employed unique attributes of the malware to detect them on the network, while the behavioral based solution observes the characteristics of programs to decide whether it is malicious or not [5]. These classifications cover various techniques such as heuristic, cloud base, model base checking, Internet of things, machine learning among others to detect malware on wireless network, but despite the success there is issues of system reliability due to adversarial perturbation method.

Adversarial perturbation is a process of manipulating the malware variants using probabilistic calculus and then attack the network. This approach has achieved great success for cyber-attack with throughput of 74.1%, detection rate of 0% for the conventional security models and hence requires urgent attention [9]. This research proposes to solve this problem using machine learning technique.

According to Ituma & Asogwa [10], machine learning is series of algorithms which have the ability to learn patterns of training dataset and then solve time series-based classification problems. These techniques will be adopted and used to develop files malware detection algorithm based on adversarial learning approach. This when achieved will provide a solution which is adaptive to perturbated malware variant features and filter them off the targeted networks.

2. Literature review

The issue of malware is a problem faced by many organizations all around the world. It doesn't come naturally like the normal infection known to human beings and some other living creatures. It is being created by malicious developers. It is a thing of joy that a counter has already being developed. The most common of those is the normal virus scan that eliminates various malware from the system that is the Panda Antivirus software. After a while, there is a change in the landscape of malware development. A non-malware based attack or the fileless malware attack was brought to life [11]. This type of attack can evade all known virus scan because it has no executable file and resides permanently in the main memory undetected making changes in the file system.

The attack using fileless malware technique aims at various vulnerabilities in the system. These vulnerabilities come from already installed application in the system which maybe the web browser, Microsoft office, PDF viewer etc. When these vulnerabilities are sighted in any of the application, it loads a script into its running memory. This is done without touching the file system [12]. These attackers have the capability of locally or remotely controlling the system with the help of the fileless malware.

According to Ref. [13], "A new genus of malware has emerged that breaks the rules of traditional detection and defence methods, unlike other breeds of malware that require the installation of software on a victim's machine, fileless malware infects a host computer dynamic memory. Fileless malware can also hijack Windows essentially turning the power of the OS against its own users by using common tools like PowerShell (which is integrated into Windows 8) for its malicious activities. Table 1presents a systematic review of the literature in this research.

3. Methods

The methods employed for the system development are data collection, data extraction, artificial neural network, activation function, training and filter-based classification.

3.1. Data collection

Fileless malware data were collected from Cyberdome as the primary source of data collection. The sample size of the data collected was 117,312 features of fileless malware. The same data was perturbated to formulate the adversarial fileless malware features used to form the second training dataset. The total sample size of data collected was 234624 attributed of fileless malware and was stored as the training dataset. The model used for the generation of the perturbated data was presented in the next section.

3.2. Generation of the fileless malware data

Today, the advancement in technology and also the increased success of penetration rate of cyber-attack has resulted to various attack techniques, but the optimization technique offers the best penetration and better throughput when compared to the rest [14]. It is a common knowledge that to train a classifier and detect malware in wireless network, a reference model parameter θ which reduced the empirical loss function for a given set of malware samples $x_1, x_2, \cdots, x_n$ is formulated as shown in equation (1);

$$\min \theta \sum_{x} \text{loss}\ (x,\ \theta)\,. \tag{1}$$

Table 1. Systematic review.

Ref.	Title	Techniques	Work done	Research gap/limitation
[15]	Analysis of Machine Learning Techniques for The Detection of Online Malware in The Cloud	Machine learning	The study used CNN, SVM, RF, K-NN, Gradient Boosted Classifier (GBC) and Gaussian Naiver Bayes (GNB) respectively to solve the problem of malware on cloud architecture	It cannot detect pertubated malware features
[16]	Integrated Malware Detection and Classification System	Hybrid approach	The study used static and dynamic malware detection techniques to develop a malware detection system	97% accuracy was achieved but pertubated malware features not considered
[9]	A machine learning based malware detection solution using texture malware perturbation method.	Malware Perturbation Method	Perturbation Features are Added to The Test Dataset Based on Gradient Descent And L-Norm Optimization Method to Attack Networks	The result showed that the machine learning algorithms (RF, CNN, and SVM) achieved detection accuracy of 0% and 74.1% throughput to the attack.
[17]	Malware classification based on probability scoring and machine learning technique	Hybrid technique	The study developed a solution which used probability threshold and spatial pyramid pooling based CNN to develop a solution which can detect malware attack.	The algorithm was trained with 174,607 samples of malware data from 63 classes of malware. The result achieved is 98.82% accuracy but pertubated malware features was not considered
[18]	Adversarial deep learning for cognitive radio security: Jamming attack and defense strategies,	Deep learning	The study developed a defense strategy against jamming attack on cognitive radio using deep neural network	The study never considered fileless malware attack
[19]	Adversarial attacks with multiple antennas against deep learning-based modulation classifiers,	Deep learning	Deep neural network was used to develop an adversarial attack detection model on multiple radio frequency antenna system	The study did not consider malware attack

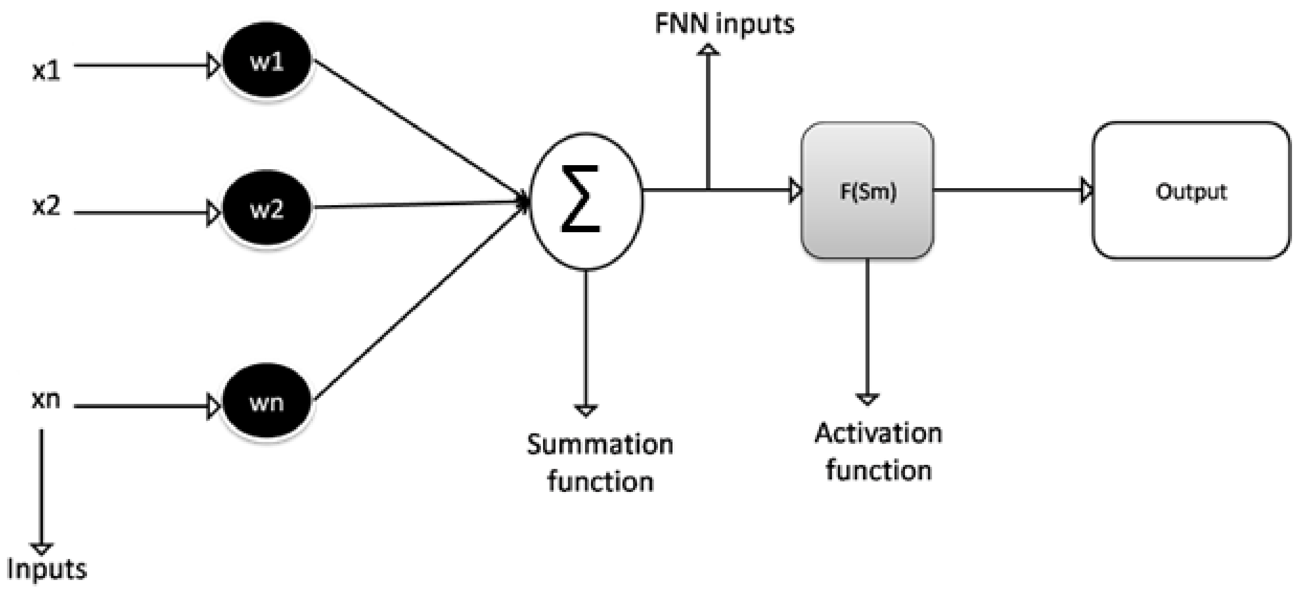

Figure 1. FFNN architectural model.

So, to cause a misclassification problem for the fixed model θ and "benign" input malware samples of x, bounded pertubation δ was introudce to ensure that the loss on x + δ is as large as possible as shown in equation (2);

$$\max \delta \ \text{loss}\ (x + \delta, \theta). \quad (2)$$

Equation (2) can be formulated applying empirical approximation technique [20] as;

$$\min \delta \sum_{x \in D} l\ (x + \delta, \theta), \quad (3)$$

where ℓ is the machine learning loss function and x∈D is the malware training dataset. The perturbated and normal fileless malware data were integrated as the training dataset and used to also formulated the targeted values.

3.3. Data extraction

Data extraction is a process of converting the malware attributes into a compact feature vector using static and dynamic analyses [20]. The static analysis involves the extraction of the malware opcodes and strings, while the dynamic analysis extracts features such as application programming interface system calls. It involves the transformation of the raw malware data collected into statistical features that can be processed while preserving the original information it contains. It aimed

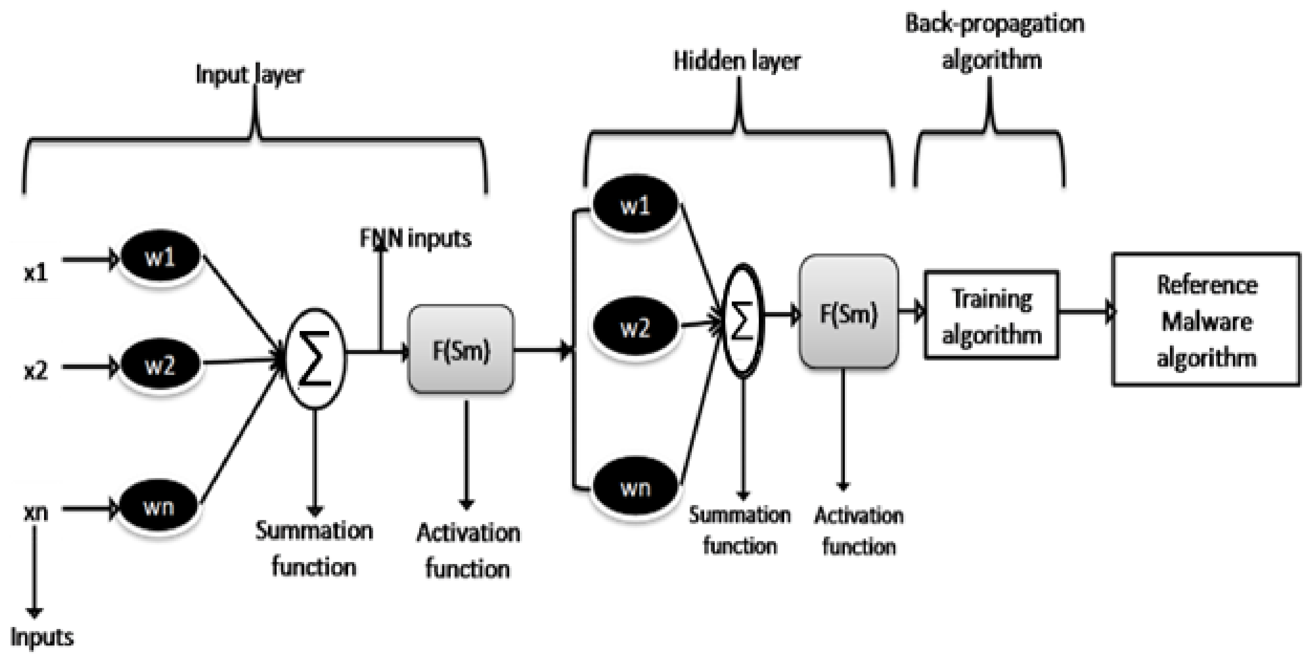

Figure 2. DNN architectural model.

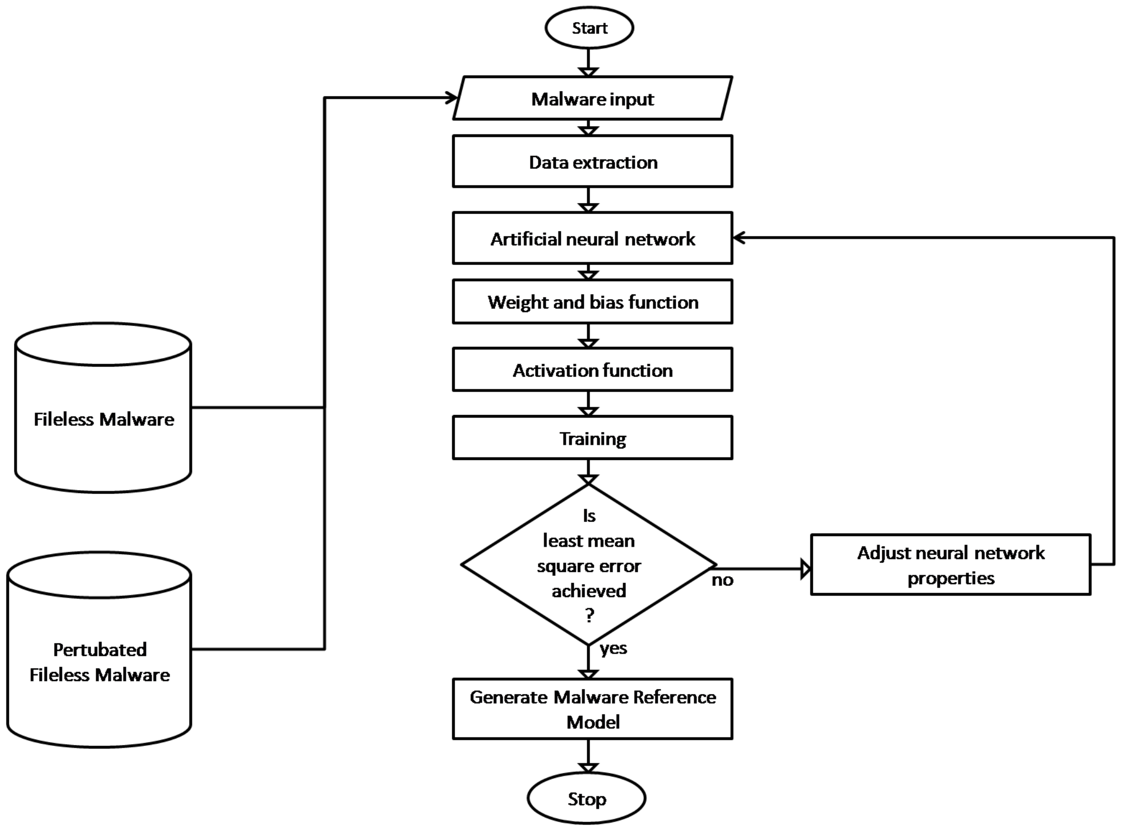
Start
Malware input
Data extraction
Artificial neural network
Weight and bias function
Activation function
Training
Is least mean square error achieved ?
no
yes
Adjust neural network properties
Generate Malware Reference Model
Stop
Fileless Malware
Pertubated Fileless Malware

Figure 3. Overview of the DNN Training Model.

at determining the subset of features which helps better differentiate between the malicious and benign files. This process plays a vital role in the accuracy and efficiency of malware detection.

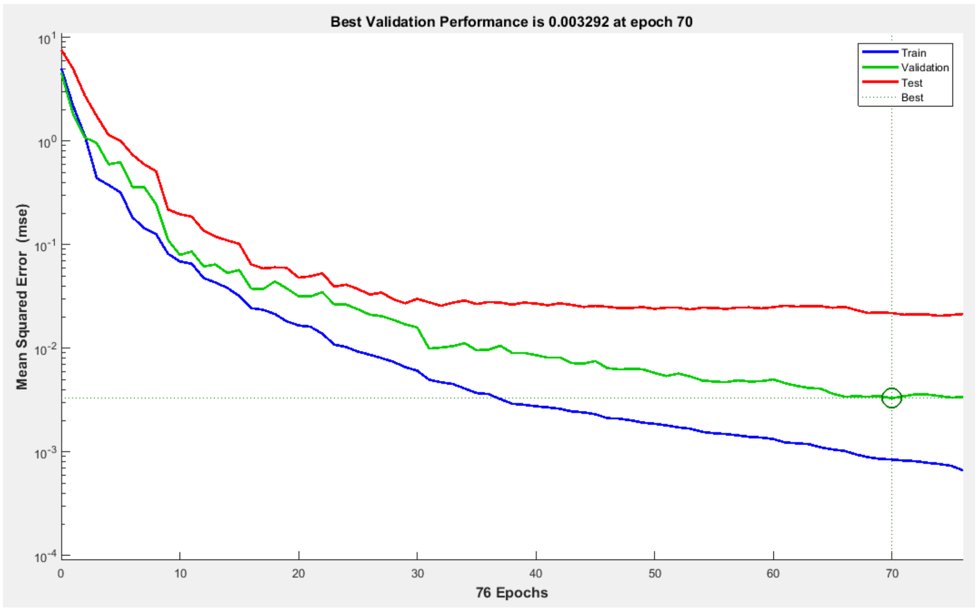

Figure 4. MSE of the DNN with GDA.

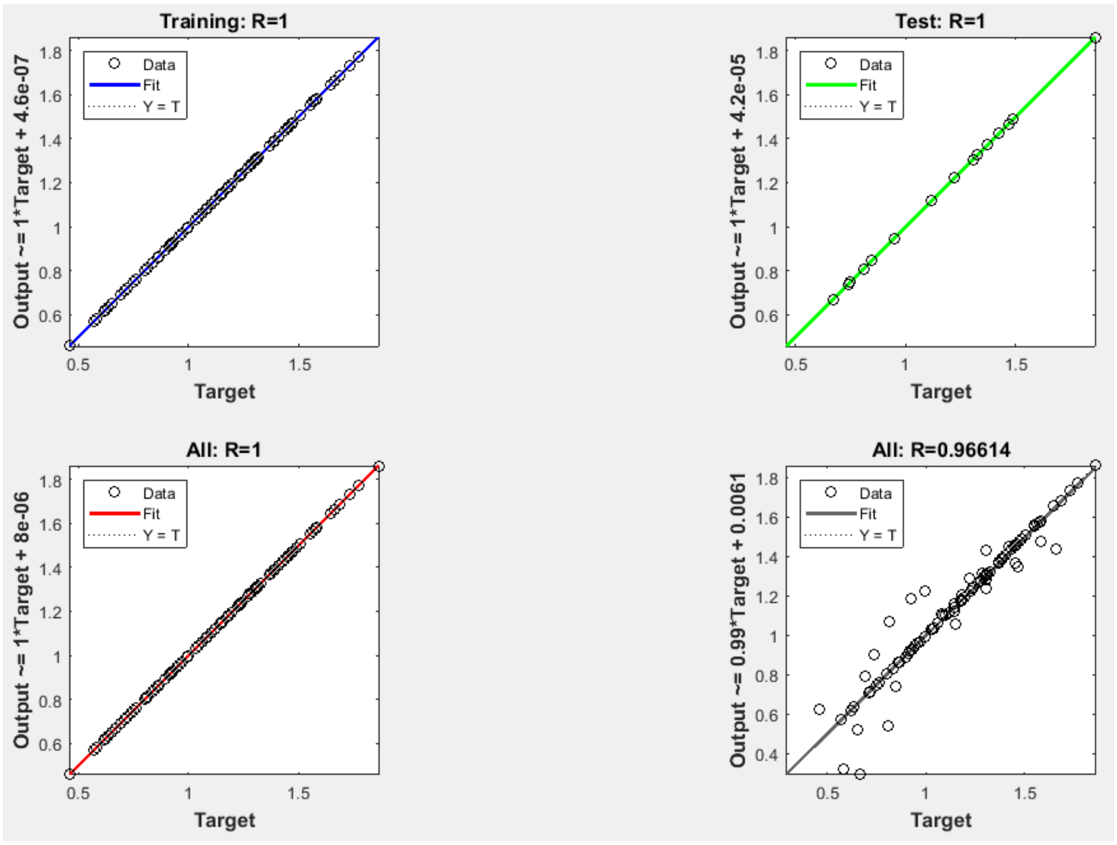

Figure 5. Regression performance.

3.4. Artificial neural network

This is the machine learning algorithm adopted for the development of the new system. The reason for adopting this algorithm over other machine learning algorithms was due to its high efficiency in solving pattern recognition and classification problem when compared with other counterparts. Artificial neural network (ANN) are biologically inspired neurons which has weights, bias and activation function with the ability to learn patterns from training set and make accurate decision. The type of neural network used is the Deep Feed Forward Neu-

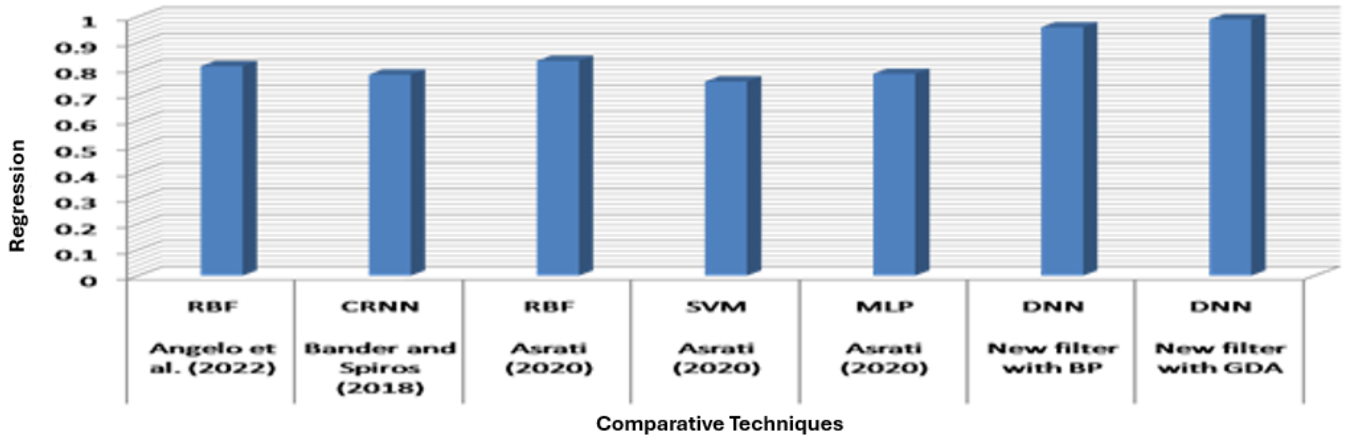

Figure 6. Comparative analysis of techniques.

Table 2. Training parameters for the DNN.

Training Parameters	Values
Maximum number of epoch to train	1000
Epoch between display	10
Maximum time to train in sec	Infinity
Validation check	6
Initial step size	0.01
Minimum performance gradient	1e-6
Cost horizon	7
Control horizon	2
Number of bias function	60
Training Iterations	76
Number of hidden layers	60
Number of weights	60
Number of input class	2

ral Network (DNN) which employed various muti configuration of the hidden layer to optimize the learning process.

3.5. Activation function

Over the years, one key element which has always been ignored for discussion in the general development of neural network algorithm is the choice of activation functions used. According to ref. [21], the right choice of activation function will guarantee efficiency in the training performance of the ANN, and improve the probability to make accurate prediction. Ref. [22] added that the conventional model of neural networks, focused on the input layers, hidden layers and output layers functionalities and will produce linear output without activation function.

The role of the activation function is to guide neural network to produce calculated output by computation of the weights and bias values via back-propagation, through the introduction of nonlinearity to ensure that output is within the ranges of [0, 1] or [-1, 1]. The activation function or the ANN used is the Rectified Linear Unit (ReLU).

3.6. Training

This involves the learning of the deep neural network using a back propagation training algorithm. The choice of this algorithm was due to its adaptive ability to measure the Mean Square Error (MSE) of the training output, feedback the errors and continue the training process until the least MSE is achieved.

3.7. Filter based classification

Classification is the process the neural network predicts the categorical labels using the learning classifier generated from the training process performed in the previous section. This will enable the network to detect the malware attack and isolate it from the network.

3.8. Challenges encountered

Gathering the fileless malware dataset presents a significant road block at this stage. These challenges include:

(a) Ephemeral nature: Fileless malware resides in memory, making it difficult to capture and analyze.
(b) No static artifacts: Unlike traditional malware, fileless malware doesn't leave behind files or static artifacts.
(c) Memory-only presence: Data is stored in memory, which is volatile and disappears after system restart.
(d) Limited visibility: Traditional security tools may not detect fileless malware due to its memory-only presence.
(e) Difficulty in capturing memory dumps: Capturing accurate memory dumps can be challenging, especially on live systems.
(f) Analyzing memory data: Understanding and analyzing memory data requires specialized skills and tools.
(g) Data volatility: Memory data can change rapidly, making it difficult to collect and analyze.
(h) System instability: Collecting data from infected systems can cause instability or crashes.

(i) Limited data availability: Fileless malware data may be scarce due to its stealthy nature.

(j) Constant evolution: Fileless malware tactics and techniques evolve rapidly, requiring continuous updates to data collection methods.

4. Development of the fileless malware filter algorithm

The malware filter was developed using the data collected and then Deep Neural Network (DNN) model. Since the perturbated data of fileless malware and standard fileless malware data have been collected and stored as the training set. The section modeled a DNN using activation function and training algorithm to generate the desired malware filter.

4.1. Modeling of the deep neural network

Deep Neural network modeling begins with a simple feed forward neural network (FFNN) architecture as shown in Figure 1 which has weight, bias, summation function and activation function.

Figure 1 present the architectural model of a simple FFNN. This model was used to develop the DNN by feeding the output of each interconnected neurons to an activation function and then feed forward to another hidden layer interconnected neural network as shown in the architectural model in Figure 2; These combination of multiple interconnected neural network layers is called DNN.

Figure 2 present the architectural model of the DNN developed from the FFNN. The model was developed by activating the output of the FFNN and then feed to the hidden layer configured with the parameters in Table 2.

4.2. Model of the ReLU activation function

According to Ref. [23], this activation function is the most used recently for deep learning algorithms due to its less computational expensive when compared with Sigmoid functions. The function can address vanishing problem recorded in Sigmoid function and is presented in the model in equation (4)

$$\begin{aligned} f(x_i) = (0,\ x_i) = \{x_i\ x_i\ < 0\ 0\ x_i\ < 0\}\ , \dot{f}\ (x_i) \\ = \{1\ x_i\ < 0\ 0\ x_i\ < 0\}. \end{aligned} \tag{4}$$

From the model, the output of the ReLU looks like linear but in fact it is nonlinear, because the output is 0 when $< 0 < 1,\ x$.

4.3. Training algorithm

The training algorithm used for the DNN development is the gradient decent back propagation algorithm. The choice of the algorithm was due to its ability to work with loss function and take the sum of squared errors during training process via the computation of gradient vectors and Jacobian matrix [24].

4.3.1. Pseudocode of the malware filter (algorithm 1)

```
i     Start
ii    Load the fileless malware dataset
iii   Extract dataset features as (m)
iv    Set the DNN configuration with table 2
v     Feed forward step (3) into step (4)
vi    Activate training algorithm (Gradient Descent Optimization)
vii   Initialize epoch as (t)
viii  Start training of neurons
ix    Check Means Square Errors and Regression at (t+1)
x     If
xi    MSE and R are ok =true
xii   Else
xiii  Adjust neural network properties
xiv   Update
xv    Do Until MSE and R are ok
xvi   Generate reference malware filter
xvii  Return
xviii Stop
```

4.4. Training model of the DNN

This training process was used to learn the neurons of the malware patterns and behaviors. This was made possible by feeding the DNN with the training data of malware. This data was extracted using static and dynamic extraction methods as identified earlier in section 3.3 into the neurons and then learn by varies the weights and bias function until least MSE error is achieved, then the output generated as the reference malware filter algorithm. The model showing how the DNN was trained is presented in Figure 3.

Figure 3 present the training model of the DNN, showing how the data collected was extracted into the DNN and trained with the back-propagation algorithm. The DNN was configured to activate the neurons with ReLU activation function before feeding to the hidden layer. The aim was to introduce nonlinearity and improve the computation efficiency, then the training algorithm was used to learn the neurons with the malware features and validate with MSE and Regression (R) in the results. Before the training process, when the training malware was loaded to the DNN using neural network toolbox, the data were divided automatically by the tool in the ratio of 70:15:15 for training, test and validation sets respectively. The training data was used to learn the neurons of the malware features; the test was used to check the learning efficiency at varying time stamp, the validation set was used to validate the training performance. These three functions sets were collectively used to generate the reference malware filter algorithm when the desired MSE and R are achieved. The pseudo code of the malware algorithm is presented as:

4.4.1. Pseudocode of the malware filter (algorithm 2)

```
1. Start
2. Load the fileless malware dataset
3. Extract dataset features as (m)
```

Table 3. Validation result of the filter with GDA.

S/N	MSE (Mu)	Regression
1	0.003292	0.9914
2	0.001254	0.9926
3	0.001155	0.9922
4	0.001358	0.9938
5	0.002333	0.9909
6	0.002715	0.9913
7	0.001752	0.9940
8	0.001753	0.9958
9	0.002214	0.9938
10	0.003051	0.9948
Average	0.002088	0.9931

4. Set the DNN configuration with table 4.1
5. Feed forward step (3) into step (4)
6. Activate training algorithm (Back propagation)
7. Initialize epoch as (t)
8. Start training of neurons
9. Check Means Square Errors and Regression at (t+1)
10. If
11. MSE and R are ok =true
12. Else
13. Adjust neural network properties
14. Update
15. Do Until MSE and R are ok
16. Generate reference malware filter
17. Return
18. Stop

4.5. *System implementation*

The filter algorithm developed was implemented using simulink and some selected toolbox embedded in the Matlab programming environment. These tools are the database toolbox used to import the training dataset, which was extracted using machine learning and statistics toolbox, and then the extracted feature vectors loaded to neural network toolbox for configuration of the DNN achieved by setting the tool using Table 2. However, during the implementation of this system, several challenges were encountered which include: occasional convergence issue encountered when several tests was conducted with back propagation algorithm, slight Overfitting arising from the use of large datasets, and computational cost involved in using large datasets with Gradient Descent Algorithm (GDA).

4.6. *Dataset characteristics*

These include:

1. Ephemeral nature: Fileless malware resides in
2. memory, making it difficult to capture and analyze.
3. No static artifacts: Unlike traditional malware, fileless malware doesn't leave behind files or static artifacts.
4. Memory-only presence: Data is stored in memory, which is volatile and disappears after system restart.

Table 4. Table comparative of techniques.

Author	Technique	Regression
Angelo *et al.* (2022)	RBF	0.8094
Bander and Spiros (2018)	CRNN	0.7777
Asrati (2020)	RBF	0.8312
Asrati (2020)	SVM	0.7531
Asrati (2020)	MLP	0.7866
New filter with BP	DNN	0.9634
New filter with GDA	DNN	0.9931

5. Limited visibility: Traditional security tools may not detect fileless malware due to its memory-only presence.
6. Difficulty in capturing memory dumps: Capturing accurate memory dumps can be challenging, especially on live systems.
7. Analyzing memory data: Understanding and analyzing memory data requires specialized skills and tools.
8. Data volatility: Memory data can change rapidly, making it difficult to collect and analyze.
9. System instability: Collecting data from infected systems can cause instability or crashes.
10. Limited data availability: Fileless malware data may be scarce due to its stealthy nature.
11. Constant evolution: Fileless malware tactics and techniques evolve rapidly, requiring continuous updates to data collection methods.

5. Results of the filter with gradient descent optimization

This section presents the result of the DNN when trained with the GDA to generate the malware filter algorithm as already presented in algorithm 3 in section 4.6. The performance of the GDA based malware filter was evaluated with the same metrics of equations (6) and (7), respectively. The MSE performance of the DN is presented in Figure 4.

Figure 4 present the new MSE of the DNN. The justification of MSE as performance evaluation parameters has already established with back-propagation. The new result of the MSE is 0.003292Mu which is good as it is approximately zero (0), indicating good training performance.

The MSE result also showed that there is a mutual relationship between the three multisets (train, test and validation) which implies that overfitting was not recorded in the training process.

The next result showed the regression performance of the DNN as shown in Figure 5.

From the result in Figure 5, it was observed that the training process was very good as overfitting was not recorded. This was because there was perfect fitting between the data in each set respectively, except in the overall result which recorded slight overfitting which was normal due to the large volume of data trained, but this did not affect the reliability of the overall regression result which is R= 0.9914 as overfitting never happened in the other three multisets. The implication of the result showed that the use of gradient descent algorithm was very good in training deep neural network to produce malware filter which is reliable.

5.1. System validation

The performance of the malware filter algorithm was validated using ten-fold cross validation techniques as presented by Ref. [25] using equation (5), with E presenting the regression values and M presenting the MSE values.

$$V = \frac{1}{10}\sum_{i-1}^{10} ME_i, \tag{5}$$

$$MSE = \frac{1}{n}\sum (y - \check{y})^2, \tag{6}$$

where y is the actual data and y̌ is predicted malware data, n is the samples size of the malware data, MSE is Mean Square Error output result of the algorithm. The other parameters choosen for the evaluation of the filter is regression. This parameter is employed to determine the relationship between dependent variable and independent or multi variable as presented in the equation (7)

$$R = f(X_i, \beta) + e_i, \tag{7}$$

where R is the dependent variable, f is the function; X_i is independent variable, β is unknown parameter and e_i is the error terms.

The regression of the filter performance generated with GDA was presented as shown in Table 3.

Table 3 present the system validation of the filter with GDA, from the result it was observed that average MSE result is 0.002088 and average R is 0.9931. The implication of these result showed that the filter developed was able to detect malware attack of all forms (i.e., perturbated) and then protect the network perfectly.

The Table 4 was used to perform a comparative analysis of the new malware filters and the existing state of the art algorithms as shown in the graph of Figure 6.

Figure 6 presented the performance of the new filter developed with the existing state of the art algorithms. From the result, it was observed that the new filter with back propagation algorithm and GDA perform better in terms of training performance.

The reason why this DNN performs better than even the deep learning in the comparative analysis was due to the use of the activation at each output of the neural network layer before feeding to the next layer, this ensures better computation performance and also the deep configuration of the neural network all contributed to the high efficiency of the new system when compared to the existing system.

6. Conclusion

This research has successfully developed a fileless malware filter which was able to detect both normal fileless malware programs and perturbated malware programs.

1. The research presented two filter algorithms with back-propagation and gradient descent respectively.
2. The result when implemented and tested showed that the use of GDA produced the most reliable filter to detect and mitigate malware attack on network.
3. The GDA was compared with other state of the art algorithms using MSE and Regression, the result showed that the new algorithm was better as it recorded highest regression values compared to the counterparts' algorithms.

Having created and validated the filter developed, the following are recommended for cyber security;

1. The fileless malware filters should be deployed to protect internet of things devices.
2. Further studies should consider other types of attack model to improve the application of this filter.

References

[1] F. O. Catak, J. Ahmed, K. Sahinbas & Z. H. Khand, "Data augmentation based malware detection using convolutional neural networks", Peerj computer science **7** (2021) e346. https://acikerisim.medipol.edu.tr/xmlui/handle/20.500.12511/6585.

[2] I. Chibueze, A. I. Ifeanyi & A. O. Chukwuemeka, "Threats and security measures on wireless local area networks", Advances in Applied Science Research **5** (2014) 4. https://www.primescholars.com/abstract/threats-and-security-measures-on-wireless-local-area-networks-89258.html.

[3] F. U. Onu, S. E. Eneji & G. Anigbogu, "The effect of object oriented programming on the implementation of biometric security system for electronic banking transactions", International Journal of Science and Research **5** (2016). https://www.ijsr.net/getabstract.php?paperid=3071601.

[4] O. C. Agwu, *A model of hybrid agent software system for combating indigeneous spam on GSM platform*, Ph.D. dissertation, Computer Science, Ebonyi State University, Abakiliki, Ebonyi State, 2016.

[5] B. Heenaa & M. Mehtre, "Advances in Malware Detection-An Overview", Computer Science Cryptography and Security, Cornell University. https://arxiv.org/abs/2104.01835.

[6] M. Graeber, "Abusing windows management instrumentation (wmi) to build a persistent, asyncronous, and fileless backdoor", Black Hat. Las Vegas, NV, USA, 2015. .

[7] T. R. Reshmi, "Information security breaches due to ransomware attacks-a systematic literature review". International Journal of Information Management Data Insights **1** (2021) 2. https://www.sciencedirect.com/science/article/pii/S2667096821000069.

[8] G. Lee, S. Shim, B. Cho, T. Kim & K. Kim, "Fileless cyberattacks: Analysis and classification", ETRI Journal **43** (2021) 2. https://onlinelibrary.wiley.com/doi/full/10.4218/etrij.2020-0086.

[9] X. Liu, J. Zhang, Y. Lin & H. Li, *ATMPA: attacking machine learning-based malware visualization detection methods via adversarial examples*, 2019 IEEE/ACM 27th International Symposium on Quality of Service (IWQoS), Phoenix, AZ, USA, June 24–25, 2019, pp. 1–10. https://doi.org/10.1145/3326285.3329073.

[10] C. Ituma & T. C. Asogwa, "The Application Of Machine Learning for Digital Recognition of Identical Twins to Support Global Crime Investigation" International Journal of Computer Science and Engineering (IJCSE) **4** (2018) 12. https://www.ijltemas.in/DigitalLibrary/Vol.7Issue12/18-25.pdf.

[11] S. Kumar, U. Dohare, K. Kumar, D. P. Dora, K. N. Qureshi & R. Kharel, "Cybersecurity measures for geocasting in vehicular cyber physical system environments", IEEE Internet of Things Journal **6** (2018) 4. https://ieeexplore.ieee.org/abstract/document/8474336/.

[12] S. M. Pontiroli & F. R. Martinez, *The tao of. net and powershell malware analysis*, Virus bulletin conference, 2015, pp. 184–196. https://www.virusbulletin.com/conference/vb2015/abstracts/tao-net-and-powershell-malware-analysis.

[13] The new breed of fileless malware and how it can be stopped with behavioural analytics and machine learning by P. Borkar. [Online]. Visited February, 02, 2020. .

[14] A. Bahramali, M. Nasr, A. Houmansadr, D. Goeckel & D. Towsley, *Robust adversarial attacks against DNN-based wireless communication systems*, ACM SIGSAC Conference on Computer and Communications Security, Republic of Korea, 2021, pp. 126–140. https://dl.acm.org/doi/abs/10.1145/3460120.3484777.

[15] J. C. Kimmell, M. Abdelsalam & M. Gupta, *Analyzing machine learning approaches for online malware detection in cloud*, 2021 IEEE International Conference on Smart Computing (SMARTCOMP), Irvine, California, USA, 2021, pp. 189–196. https://ieeexplore.ieee.org/abstract/document/9556309/.

[16] T. Ronghua, *An integrated malware detection and classification system*, PhD. dissertation, Deakin University, Australia, 2011. .

[17] D. Xue, J. Li, T. Lv,W. Wu & J. Wang, "Malware classification using probability scoring and machine learning", IEEE Access **7** (2019) 91641. https://ieeexplore.ieee.org/abstract/document/8758215.

[18] Y. Shi, Y. E. Sagduyu, T. Erpek, K. Davaslioglu, Z. Lu & J. H. Li, *Adversarial deep learning for cognitive radio security: Jamming attack and defense strategies*, IEEE international conference on communications workshops (ICC Workshops), Kansas City, MO, USA, 2018, pp. 1–6. https://ieeexplore.ieee.org/abstract/document/8403655.

[19] B. Kim, Y. E. Sagduyu, T. Erpek, K. Davaslioglu & S. Ulukus, *Adversarial attacks with multiple antennas against deep learning-based modulation classifiers*, IEEE Globecom Workshops (GC Wkshps), Taipei, Taiwan, 2020, pp. 1-6. https://ieeexplore.ieee.org/abstract/document/9367473/.

[20] D. Birkić, A. Primuzak & P. Silvija, "Integral quality management in a Continental tourism destination: shown on the example of Pozega Slavonia County the City of Pozega", In *8. medjunarodni znanstveni simpozij: Gospodarstvo istocne Hrvatske–vizija i razvoj*, 8th International Scientific Symposium: Economy of Eastern Croatia–Vision and Growth, Croatia, 2019, pp. 663–680. https://www.croris.hr/crosbi/publikacija/prilog-skup/687659.

[21] V. Nair & G. E. Hinton, *Rectified linear units improve restricted Boltzmann machines*, 27th International Conference on Machine Learning (ICML10), Haifa, Israel, 2010, pp. 807–814. https://dl.acm.org/doi/abs/10.5555/3104322.3104425.

[22] J. Saxe & K. Berlin, *Deep neural network based malware detection using two dimensional binary program features*, 10th international conference on malicious and unwanted software (MALWARE), ajardo, PR, USA, 2015, pp. 11–20. https://ieeexplore.ieee.org/document/7413680/.

[23] J. Feng & S. Lu, "Performance analysis of various activation functions in artificial neural networks", Journal of physics: conference series **1237** (2019) 2. https://iopscience.iop.org/article/10.1088/1742-6596/1237/2/022030/meta.

[24] V. Nair & G. E. Hinton, *Rectified linear units improve restricted boltzmann machines*, Proceedings of the 27th international conference on machine learning (ICML-10), Haifa, Israel, 2010, pp. 807–814. https://dl.acm.org/doi/abs/10.5555/3104322.3104425.

[25] 5 Algorithms to Train Neural Network by Q. Alberto. (2018). Neural Designer; Data Science and Machine learning Blog; ARTELNICS. https://www.neuraldesigner.com/learning/.