EEJOURNALS OPEN ACCESS
EURASIAN EXPERIMENT JOURNAL OF ENGINEERING(EEJE) ISSN: 2992-409X
©EEJE Publications Volume 4 Issue 1 2023

Development of a multi-level security model for the log management system using machine learning technique

[1]Emeka Okorie*, [1]Iloka. B. C., [1]Anyaragbu Hope, [1]Okoh. C. C., [1]Ejikeme. A., [1]Dom-anyanwu. C. P., [1]Chukwura. S. and [2]Nwaukwa Johnwendy

[1]Department of Computer Science Tansian University Umunya Anambra State, Nigeria.
[2]Department of Computer Science. Veritas University Abuja.

ABSTRACT

Recently the use of Machine Learning (ML) has gain increased attention towards cyber security. ML is a branch of artificial intelligence which employed series of mathematical algorithms to learn from data and solve regression or pattern recognition problems. This solution was used to solve the problem of cloud intrusion detection in, but despite the success flood attack was never considered. Flood attack is a type of denial of service attack which has threatened the integrity of cloud based infrastructures over the years and has remained a major challenge. Developing a multi-level intrusion detection algorithm employs machine learning technique to monitor, detect and control flood attack on targeted cloud based infrastructures management system using machine learning technique. Cloud computing technology provides human many advantages such as economical cost reduction and effective of IT based resources. Unfortunately the expansion of the technology equally expands its vulnerabilities to threats and has remained a very big challenge over the years. This problem was addressed in this research developing a multi-level intrusion detection system for the security of cloud log management server. The solution employed ML which used neural network to develop an intelligent security model which protects the targeted server against IP flood and other forms of denial of service attack. The solution was implemented with Simulin*k and evaluated. The result showed that the security model developed was able to detect and control threats from intruding into the server.

Keywords: Development, multi-level, security, model, machine and learning technique

SYSTEM DESIGN AND SYSTEM IMPLEMENTATION

System Design

This presents the problem formulation, modelling of the multi-level intrusion detection system and the complete system modelling.

Modelling of the problem formulation

The problem formulation considered the IP flood attack which is a form of denial of service attack used by hackers to target cloud based infrastructures and initiate ransomeware. The architectural model in figure 1 presented the workflow of the attack penetration.

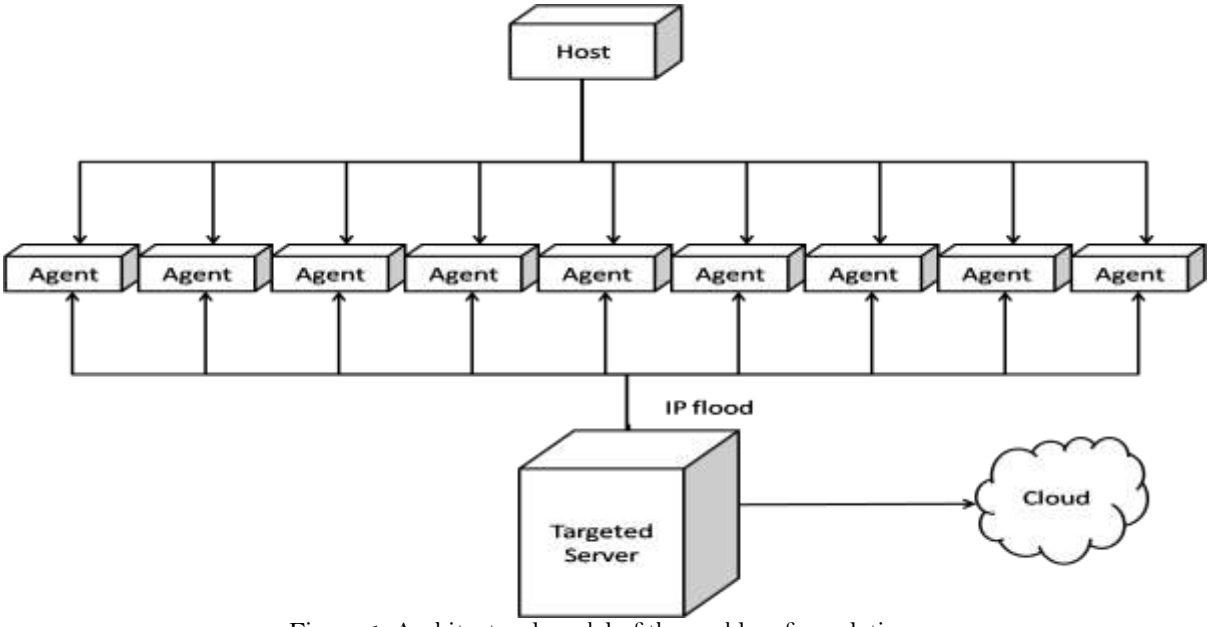

Figure 1: Architectural model of the problem formulation

In the figure 1, the attacker (host) created multiple agents which help in the distribution of the threat simultaneous to the targeted host. When this happens the host server is overwhelmed and then suffers congestion problem and eventual shutdown [1-3].

Modelling of the neural network algorithm

The neural network algorithm was formulated from a single neuron as shown in the architectural model in figure 2;

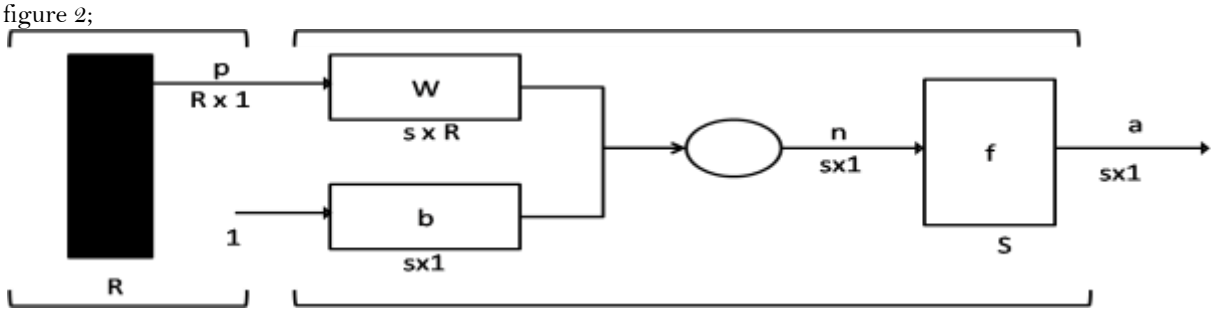

Figure 2: model of a single neuron

Where a is the output of the neuron after activation, p is the input vector, W is the neuron weight, b is the bias function of the neuron, n is scalar output, R is the number of elements in the input vector, s is the number of neurons in the input layer. The neuron in figure 4.2 was interconnected to form the multi layered neural network architecture in figure 3;

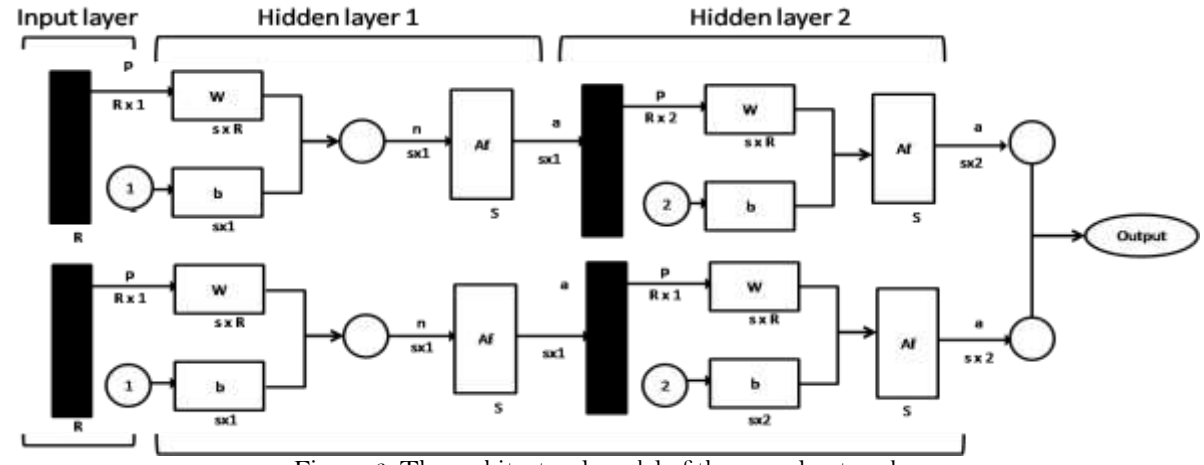

Figure 3: The architectural model of the neural network

The figure 3 presented the architectural model of the neural network algorithm which was configured to train the data collected from the log server. The neural network algorithm was developed with specified number of neurons based on the data attributes of the log server input , Rectified Linear Unit (ReLU) as the activation function (Af) and back propagation algorithm. The ANN algorithm was loaded with the data and then trained to generate the detection algorithm. The neural network training model was presented in figure 4;

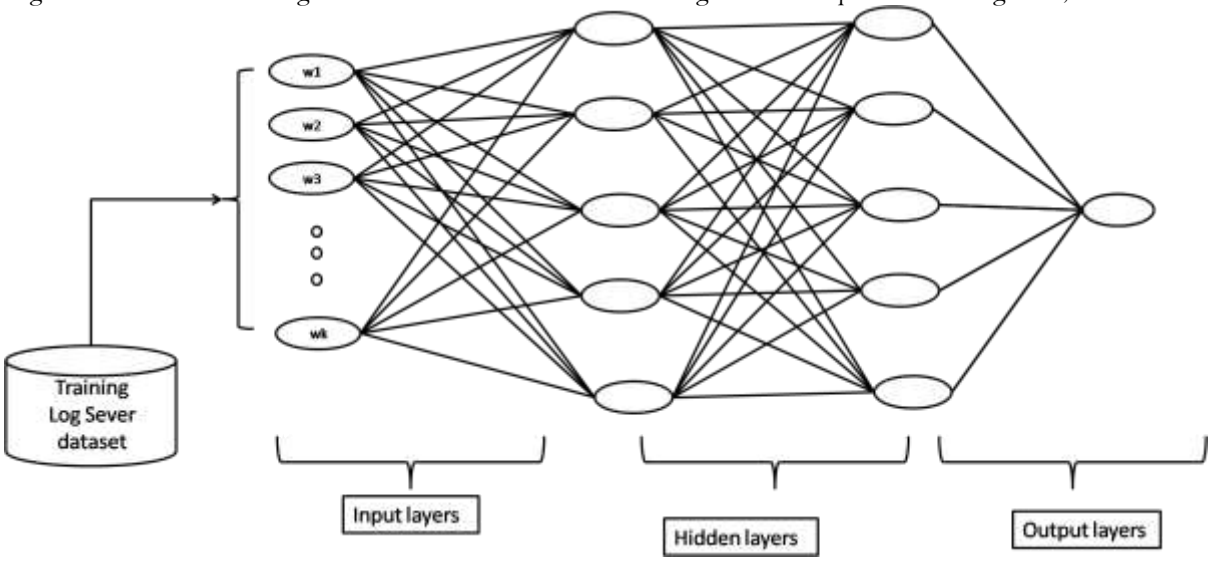

Figure 4: The neural network training model

The figure 4 presented the neural network model which was used to train the data collected to generate the detection algorithm. Before the training, the neural network splits the training data into training, test and validation set. The training was used to learn the neurons of the log patterns on the server, the testing was used to tested the regression performance and then the validation was used to ensure reliability in result. During the training process the neurons are adjusted to learn the log in patterns on the server as various time logs, iteratively using a back-propagation training algorithm as shown in figure 4.5, until a constant output was achieved [4-6].

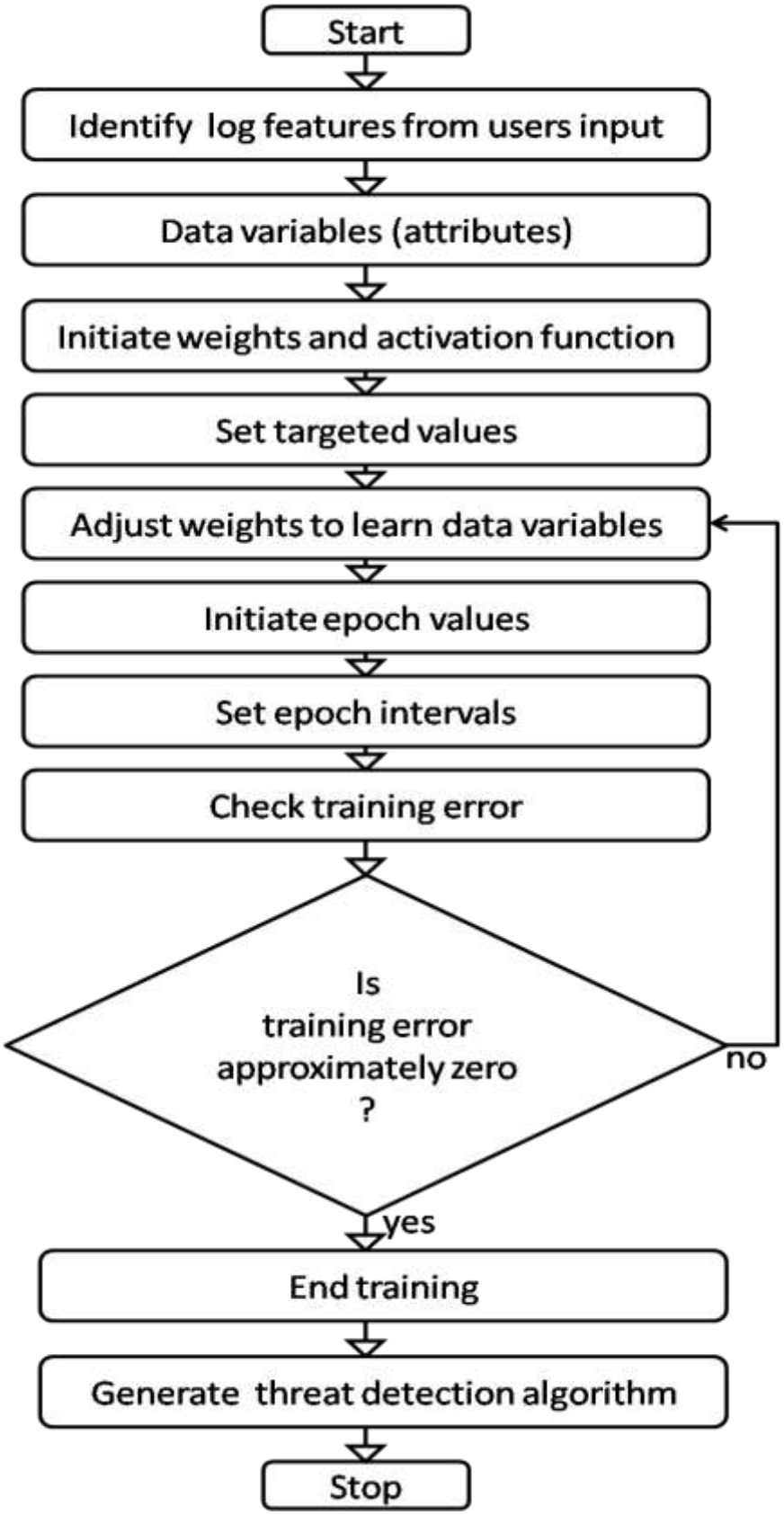

Figure 5: The back-propagation training algorithm

The figure 5 presented the training algorithm which enables the neuron to learn. During this learning process, the output is evaluated for leas mean square error and also regression capabilities to meet the targeted values. When this is aim not achieved, then it was feedback and the neurons are adjusted and retrained again until the least means square error is achieved. Then the training stops and then algorithm generated as shown in the flow chart of figure 6;

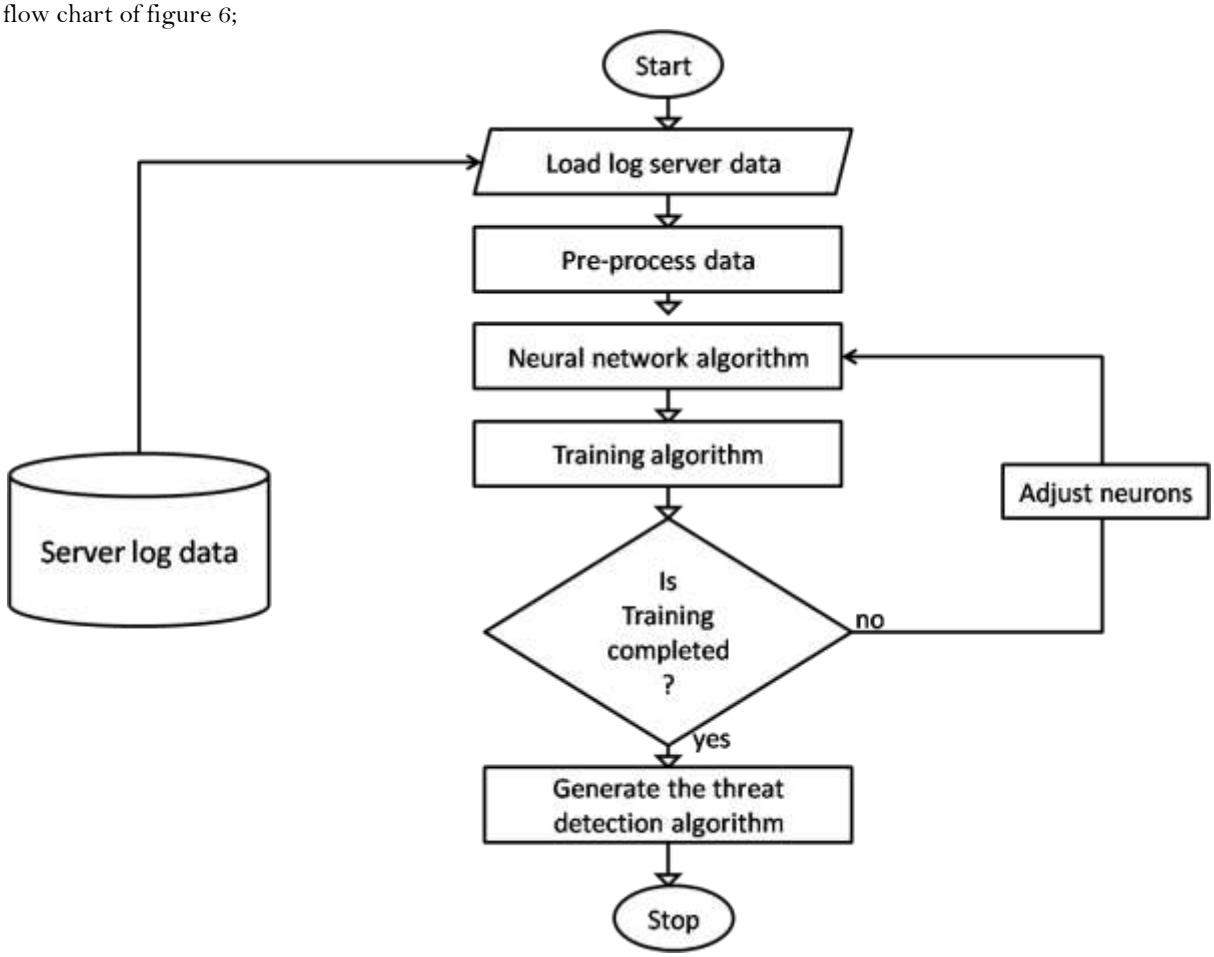

Figure 6: Flow chart of the threat detection algorithm

The threat detection pseudocode (Algorithm 1)

1. *Start*
2. *Load log server data*
3. *Divide data into training and target set*
4. *Configure neural network*
5. *Initiate training algorithm*
6. *Initiate activation function*
7. *Set training epoch values and interval*
8. *Train neural network*
9. *If*
10. *Training is completed = true*
11. *Generate threat detection algorithm*
12. *Else*
13. *Return to training algorithm*
14. *End if*
15. *End*

The model of the threat control algorithm

The algorithm developed in the figure 6 presented the threat detection flow chart. This was used for the monitoring and detection of threat log to the server. However, to control the problem so as to prevent it from passing through the cloud server the flow chart in figure flow chart in figure 7;

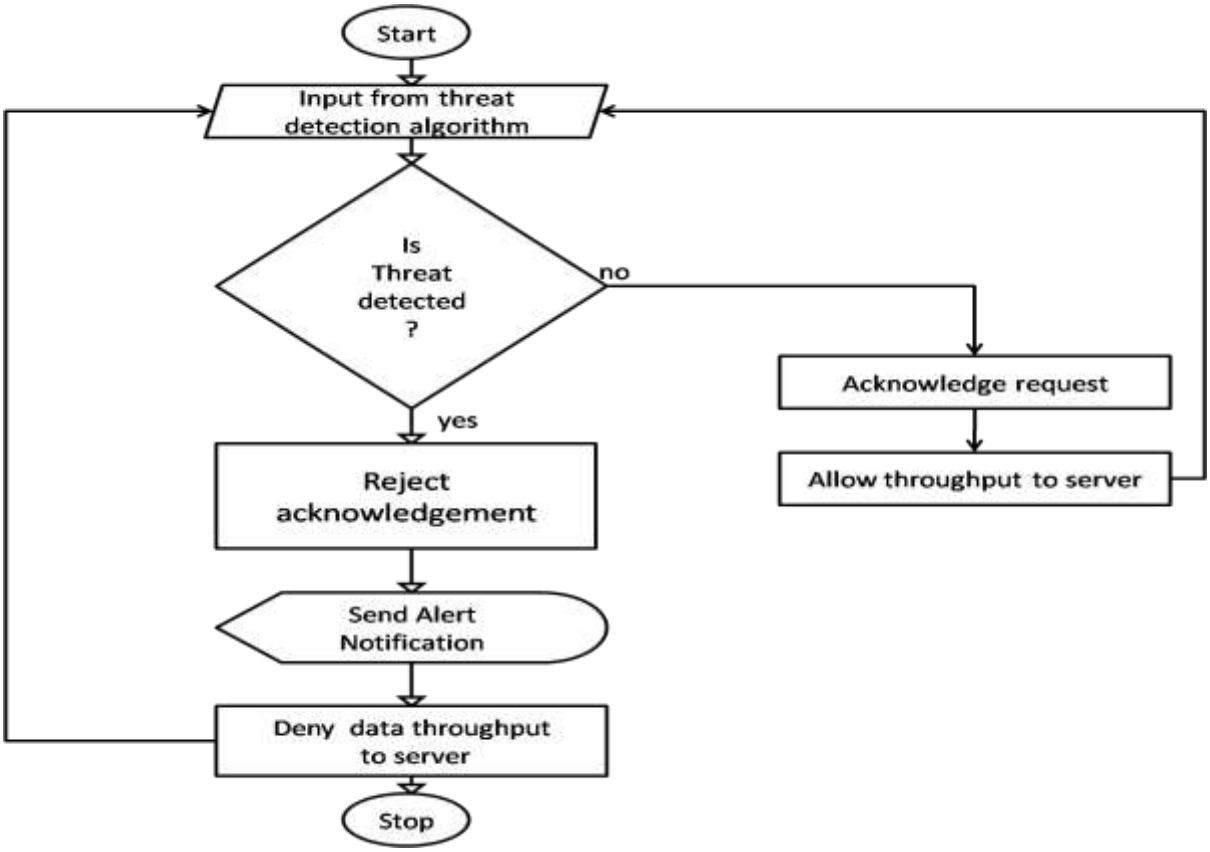

Figure 7: The flow chart of the threat control algorithm

The figure 7 presented the threat control algorithm which identified the output from the monitoring and detection algorithm and then if threat was detected isolate it from the server to prevent shutdown. The pseudopodium of the control algorithm is presented as;

1. *Start*
2. *Identify input from algorithm 1*
3. *If*
4. *Threat is detected = true*
5. *Log Server Rejects acknowledgement of packet*
6. *Send alert notification*
7. *Deny data throughput to cloud*
8. *Else*
9. *Acknowledge request*
10. *Allow throughput to cloud*
11. *End if*
12. *Return to algorithm 1*
13. *End*

Development of the multi level intrusion detection system

The model of the multi level intrusion detection system was developed to monitor, detect and control the penetration of threat on the log manager. This was achieved using the threat detection algorithm and the control algorithm to develop a multi level intrusion detection system. The model of the new system is presented in the flow chart of figure 8;

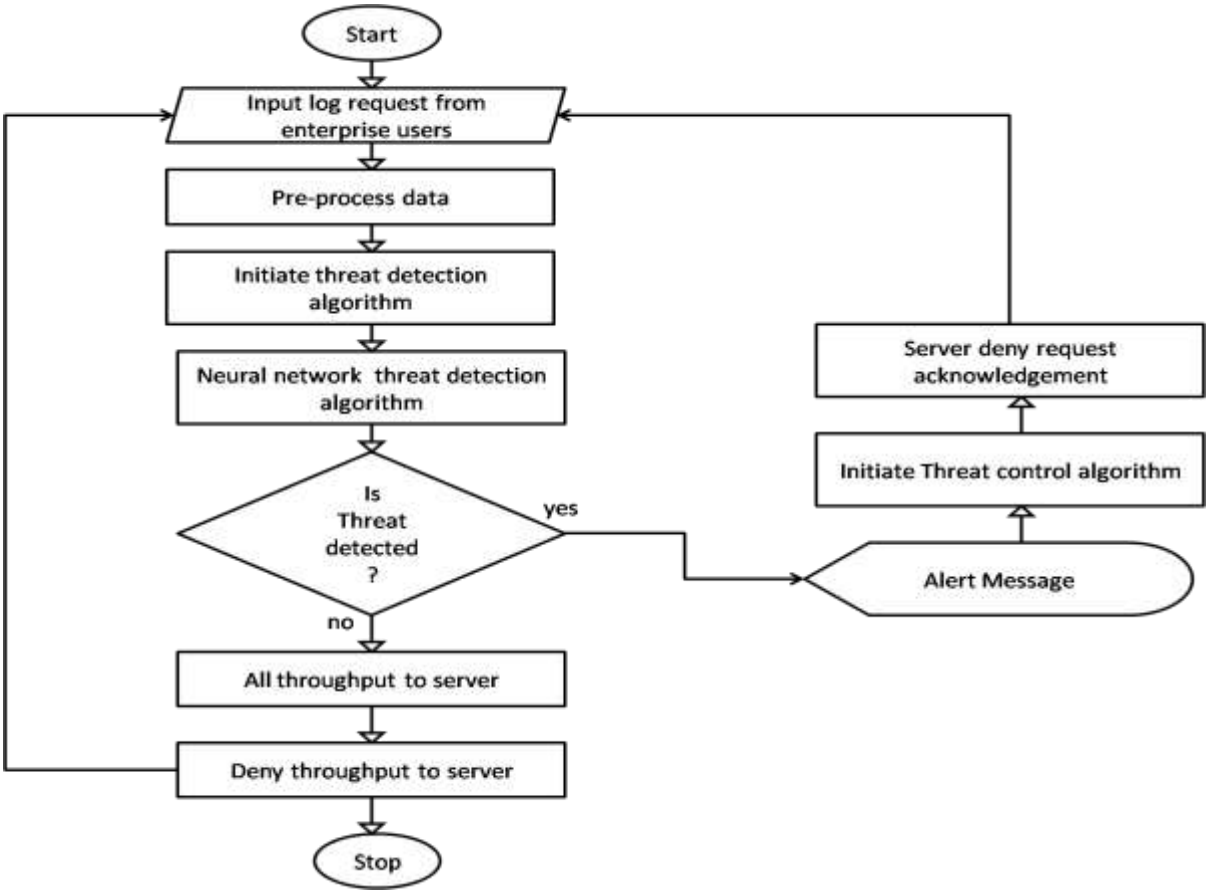

Figure 8: flow chart of the multi level intrusion detection system

The flow chart in figure 8 presented the workflow of the intrusion detection system, showing the logical relationships between the various modules which interacted to achieve the complete system. The data input from the user enterprises as logged into the server, the adopted processing algorithm removed noise with image formats which is common with such data and then the algorithm 1 was used to train and detect threat for control via access denial to the target using the threat control algorithm, else when threat was not detected, throughput is allowed to the server [8-9].

Sequence Modelling of the Multi intrusion detection system

The sequence modelling approach is a Universal Modelling Language (UML) diagram which used objects to develop the sequence modelling for the threat detection and control algorithms, thus showing how the various process interact to detect threat and prevent it from the server and at the same time allow normal enterprise log request to go through as it has no threat attributes. The sequence modelling will presented the model of the algorithm for threat detection and also the model of the algorithm to facilitate log management for normal enterprise users [10-11].

Sequence diagram for threat detection

The figure 9 explains the sequence diagram of the threat detection algorithm using objects such as the attacker node, pre-processing, threat detection algorithm, training, threat control algorithm, and the server. From the operational sequence, the attacker sends the IP flood to the targeted server, the data are identified and processed before training by the threat detection algorithm and if threat is detected, the threat control algorithm was used to prevent it access to the cloud.

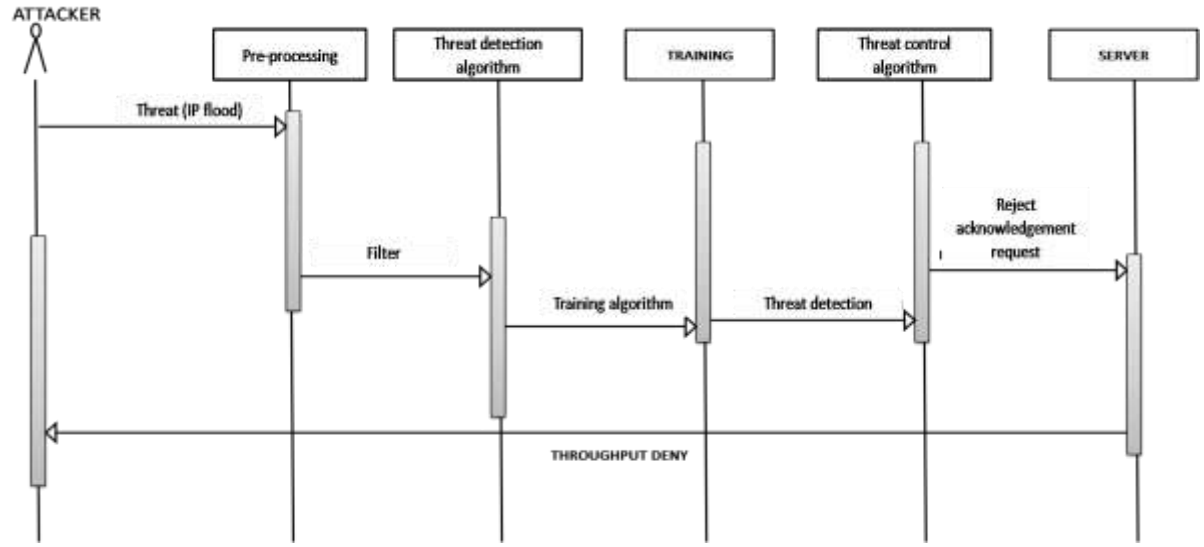

Figure 9: Sequence diagram for normal program file data transfer

Sequence diagram of the threat algorithm on normal packet

The figure 10 explains the sequence diagram of the threat detection algorithm using objects such as the enterprise node, pre-processing, threat detection algorithm, training, threat control algorithm, and the server. From the operational sequence, the enterprise sends login data to the server, the data are identified and processed before training by the threat detection algorithm and if threat is not detected, throughput is allowed to the cloud.

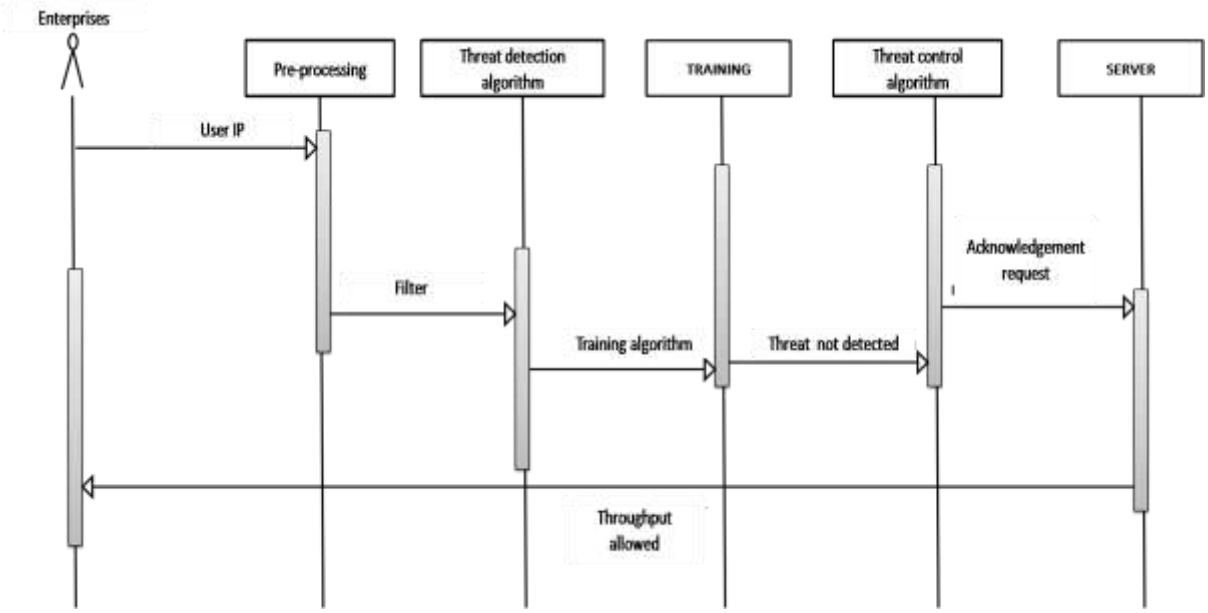

Figure 10: Sequence modelling of normal data log service

System Implementation

The system designed was implemented with system identification toolbox, statistics and machine learning toolbox, optimization toolbox, neural network toolbox, communication toolbox and Simulink. The system identification toolbox was used by the neural network to load the training dataset as shown in the dataset wizard in figure 11;

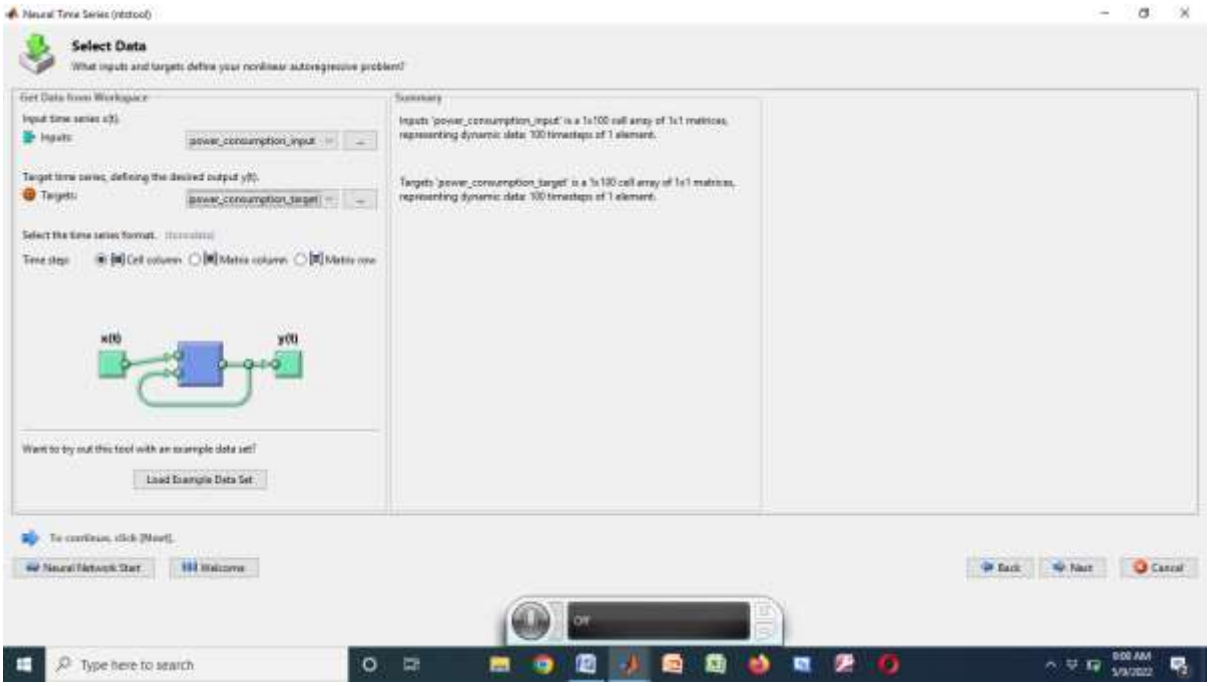

Figure 11: The ANN data import wizard tool

The figure 11 showed how the dataset was loaded into the neural network model for configuration as shown in the figure 12;

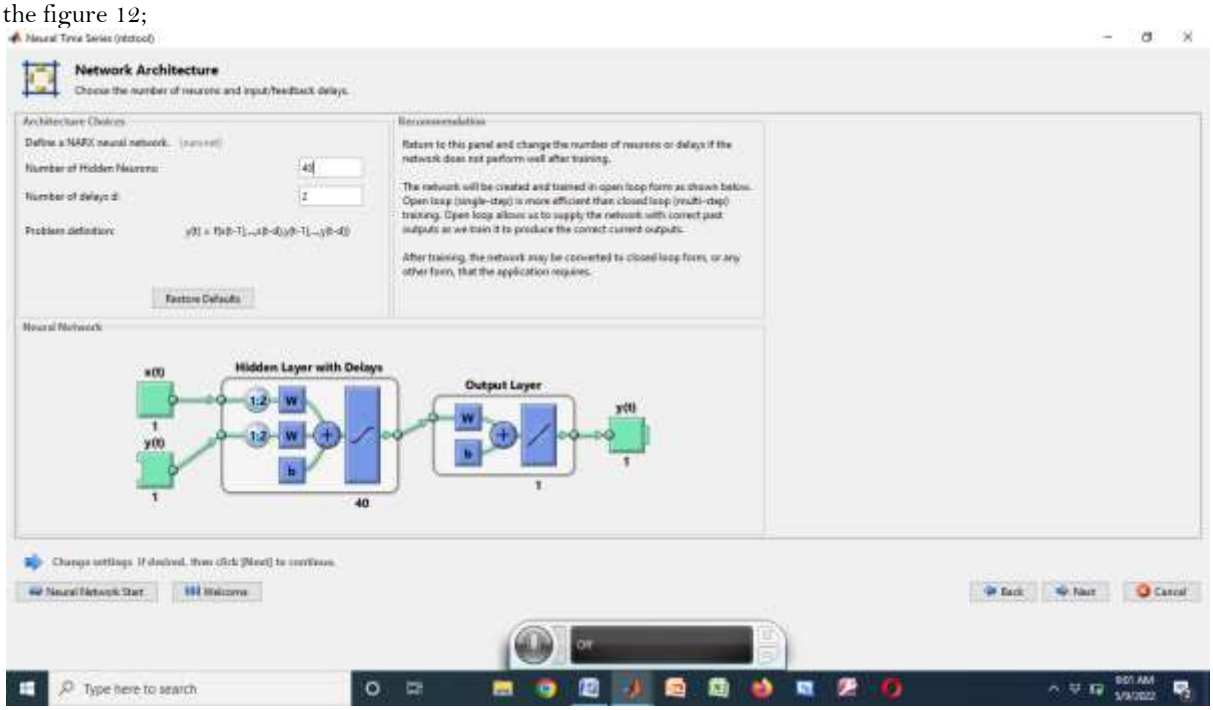

Figure 12: The neural network configuration tool

The figure 12 showed how the neural network was configured with the input of the number of hidden layers and the delay values. The next result showed where the training algorithm was selected and the training process taking place as in figure 13;

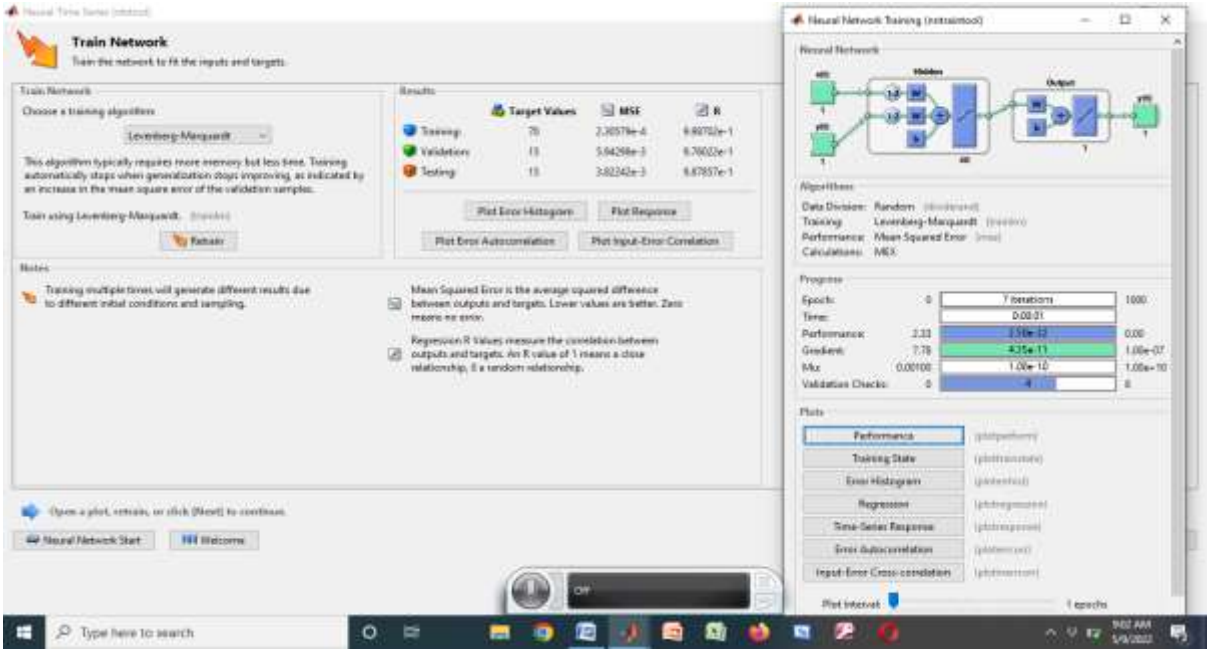

Figure 13: The neural network training tool

The figure 13 showed the neural network training tool used to train the neurons of the data collected to generate the threat detection algorithm. This algorithm was married with the threat control algorithm using basic Matlab script programming.

Contribution to Knowledge

i. A threat detection algorithm was developed with neural network
ii. A threat control algorithm was developed
iii. A multi level intrusion detection system was developed for cloud log management server

CONCLUSION

This research has successfully developed a multi level intrusion detection system for cloud based log management system using machine learning technique. The researcher conducted an empirical study of a cloud collaborative platform to identify its vulnerabilities and then develop a neural network based security algorithm for the detection of intrusion. Rule based approach was used to control the detected threat so as to isolate it from the cloud. The two security algorithms developed were married as multi level intrusion detection system and implemented with Simulink. The result after testing and validation recorded MSE of 2.47e-05Mu and R= 0.99445. The implication of the results showed that the new algorithm developed was able to correctly monitor, detect and prevent threat penetration to cloud based server. A comparative analysis was also conducted with other threat detection algorithms and from the result, it was observed that the performance of the new system was better due to its multi layered configuration of neurons to enhance data processing and computation in the hidden layers, then right choice of activation used, training algorithm adopted and also the quality of data used to train the neural network and achieve the threat detection algorithm. These features of the new algorithm make it to standout from the others with better performance.

REFERENCES

1. Axelsson S, (1999). Research in Intrusion-Detection Systems: A Survey,tech. report TR-98-17, Dept. Computer Eng.,Chalmers Univ. of Technology, 1999.
2. Bakshi A., Yogesh B, (2010)"Securing cloud from DDOS Attacks using Intrusion Detection System in Virtual Machine", Second International Conference on Communication Software and Networks, pp. 260-264.
3. Bharadwaja S., Sun W., Niamat M., Shen F., (2011) "Collabra: A Xen Hypervisor based Collaborative Intrusion Detection System", Eighth International Conference on Information Technology: New Generations, pp. 695-700.
4. Doyen Sahoo, Chenghao Liu, and Steven CH Hoi. (2019) malicious url detection using machine learning: A survey. arXiv preprint arXiv:1701.07179.
5. ENISA. (2020). ENISA Threat Landscape 2020 – Web-based Attacks. DOI: 10.2824/552242 [Online] Available at: https://www.enisa.europa.eu/publications/webbased-attacks; Retrieved on 2/19/2022

6. Hodo, X. Bellekens, A. Hamilton, P.-L. Dubouilh, E. Iorkyase, C. Tachtatzis, and R. Atkinson, "Threat analysis of IoT networks Using Artificial Neural Network Intrusion Detection System," in 2016 3rd International Symposium on Networks, Computers and Communications (ISNCC), 2016, pp. 1–6.
7. Jolera. (2020). 3 Ways AI Prevents Phishing Attacks. [Online] Available at: https://www.jolera.com/3-ways-ai-prevents-phishing-attacks/; Retrieved on 3/2/2022
8. Phillip George Efthimion, Scott Payne, and Nicholas Proferes. Supervised machine learning bot detection techniques to identify social twitter bots. SMU Data Science Review, 1(2):5, 2018.
9. Ramgovind S, Eloff MM, and Smith E (2010). Research on The Management of Security in Cloud computing appear in 2010 IEEE
10. U.S. Patent (1994). A network [...] is shown schematically as a cloud", U.S. Patent 5,485,455, column 17, line 22, filed Jan 28, 1994. Retrieved from http://www. google. com/ patents?vid=5790548
11. Uday Trivedi and Munal Patel. A fully automated deep packet inspection veri_cation system with machine learning. In 2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), pages 16. IEEE, 2016.

Emeka Okorie, Iloka. B. C., Anyaragbu Hope, Okoh. C. C., Ejikeme. A., Dom-anyanwu. C. P., Chukwura. S. and Nwaukwa Johnwendy (2023). Development of a multi-level security model for the log management system using machine learning technique *EURASIAN EXPERIMENT JOURNAL OF ENGINEERING(EEJE)* 4(1):121-131