ABUAD Journal of Engineering Research and Development (AJERD)
ISSN (online): 2645-2685; ISSN (print): 2756-6811

Volume 7, Issue 2, 297-307

Modelling of Cyber Attack Detection and Response System for 5G Network Using Machine Learning Technique

Anthony KWUBEGHARI[1], Lucy Ifeyinwa EZIGBO[1], Francis Amaechi OKOYE[2]

[1]*Computer Engineering, Enugu State University of Science and Technology, Agbani, Enugu State, Nigeria*
kwubeghari@gmail.com/lucyezigbo752@gmail.com

[2]*Computer Engineering, Enugu State University of Science and Technology, Agbani, Enugu State, Nigeria*
francisced@esut.edu.ng

Corresponding Author: kwubeghari@gmail.com, +2348064238671
Date Submitted: 24/07/2024
Date Accepted: 01/09/2024
Date Published: 15/09/2024

***Abstract**: The rapid increase in the adoption of 5G networks has revolutionized communication technologies, enabling high-speed data transmission and connectivity across various domains. However, the advent of 5G technology comes with an increased risk of cyber-attacks and security breaches, necessitating the development of robust defence mechanisms to safeguard network infrastructure and mitigate potential threats. The work presents a novel approach for modelling a cyber-attack response system tailored specifically for 5G networks, leveraging machine learning techniques to enhance threat detection and response capabilities. The study introduced innovative methodologies, including the integration of standard backpropagation and dropout regularization technique. Furthermore, an intelligent cyber threat classification model that proactively detects and mitigates malware threats in 5G networks was developed. Additionally, a comprehensive cyber-attack response model designed to isolate threats from the network infrastructure and mitigate potential security risks was formulated. The result of testing the response algorithm with simulation, and considering quality of service such as throughput, latency and packet loss, showed 80.05%, 24.9ms and 4.09% respectively. During system integration of the model on 5G network with stimulated malware, the throughput reported 71.81%. Also, packet loss reported loss rate of 23.18%, while latency reported 178.98ms. Our findings contribute to the advancement of cybersecurity in 5G environments and lay the foundation for the development of robust cyber defence systems to safeguard critical network infrastructure against emerging threats.*

***Keywords:** Cyber Threat Response Algorithm (CTRA), Dropout Algorithm, Back Propagation, Machine Learning, Artificial Neural Network (ANN)*

1. INTRODUCTION

The evolution of 5G technology, as the latest generation of mobile networks, has offered significant upgrades over previous generations like 2G, 3G, 4G/LTE-A. Some of the improvements include faster routing speeds during upstream and downstream services, lower latency, improved bandwidth size, enabling new applications like the Internet of Things (IoT), and improved reliability, among others [1]. However, there are also issues that continue to threaten the overall quality of service in 5G networks.

Today, 5G networks face many challenges that affect the provision of high-quality service. Some of these challenges include network congestion, interference, security vulnerabilities, and high deployment costs, which result in low routing speeds, poor coverage, reduced reliability, poor throughput, vulnerability to cyber-attacks, and overall degraded performance that impacts the user experience [2]. However, Casillas, et al. and Maksim, et al [3, 4] submitted that the software-dependent nature of 5G has triggered a paradigm shift from conventional physical network resources to dynamic virtualization, software-defined networking, and cloud computing, and has provided many opportunities for cyber threats. Obodoeze and Francis [5] further posited that the application layer of the network architecture makes it vulnerable to attack due to the large interconnectivity of things with complex applications, while Neha, et al [2] opined that potential weaknesses in network architecture, device vulnerabilities, supply chain risks, all contribute and create opportunities for potential threats, and increase the overall risk of security breaches. This as a result makes study on 5G a research hotspot and presented the need for Cyber Attack Response System (CARS).

According to Mozo, et al [6], CARS is a form of emergency response system to an isolated attack, or a multi-pronged attack on 5G powered enterprise network infrastructures. Over the years, various techniques to solve cyber-attack problem have been developed, however Sultana et al. and Mozo, et al [6, 7] posited that the use of artificial intelligence approach has dominated the studies as the most effective tool so far.

To start with, Ogbeta and Lois [8] used neural network for the isolation of Distributive Denial of Service (DDOS) attack and recorded 0.978 regression score, while Ogbuanya and Eke [9] applied neural network for the isolation of blackhole and reported 84.45 accuracy; similarly, Oduah and Olofin [10] applied neural network for the security of cloud log management system and reported 0.9945 regression score. Despite the success, Obodoeze and Francis [5] argued that wireless network are not limited to individual threats as identified in the studies, but suffers varieties of attack which includes wormhole, virus, botnets, etc.

In Imanbayev, et al [1], multiple attack dataset were considered with overall 84 attributes and then applied to train Logistic Regression (LR), Random Forest (RF), and Gradient Boosting (GB) algorithms for cyber threat detection system. The result reported GB as the best with 99.3% success rate. However, similar success may not be achieved in reality, as the two datasets did not capture all major cyber-attack scenarios for 5G.

Maksim, et al [4] used machine learning and physical experiment to propose a CARS considering 15 attack models. This work is very interesting as multiple threat were considered for the study; however, it is not clear the exact machine learning algorithm adopted; furthermore, it is not clear how the models were validated. Therefore, while significant research has been presented on 5G CARS, most of the studies are limited to few threat models and also it was observed that generally there is no comprehensive security solution that addresses multiple security threats in 5G networks. Hence, there is need for more research to develop robust security mechanisms for 5G networks that can effectively protect against various network threats. Hence, this paper presents the application of machine learning technique for cyber-attack detection and response system for 5G network.

2. RESEARCH METHODOLOGY

The general methodology used for the research is Agile because it is an adaptive approach which allows the integration of interdisciplinary research methods [11]. The realization of the Cyber Attack Response System (CARS) involves a two-layered approach which first modelled an Intelligent Cyber Threat Classification (ICTC) algorithm using artificial neural network, then the second layer which is a cyber threat response algorithm utilized to isolate the threat from the network. These two algorithms were integrated as the CARS model for the security of 5G network.

The highlight of the research was centred on addressing the issues of over-fitting which occurred during the training of the neural network to generate the ICTC model. This was achieved using dropout algorithm [12, 13]. The CARS was tested through simulation and then integrated on a case study 5G network.The results were analysed through statistical method of system analysis. The method involved in realizing the CARS were discussed in sequence from subsection 2.1 to subsection 2.9 as follows:

2.1 Data Collection

The primary data collection focused on the network characterization of 5G network facilityof ICT department of NTA headquarters, the secondary data collection used here provided the network threat data used for the study. The source of the data collection is the Institute of Electrical Electronics Engineering (IEEE) Dataport [14] which is an open repository for studies. The sample size of the data collected is 125871 samples consisting 41 features of threats across 22 threat classes which are Back, Buffer_overflow, FTP_write, Guess_password, IMAP, Ipsweep, Land, Load_module, Multihop, Neptune, Nmap, Perl, Phf, Pod, Portsweep, Rootkit, Satan, Smurf, Spy, Teardrop, Warezcinet, Warezmaster and normal packet.

2.2 Artificial Neural Network Modelling

The neural network model utilized for the study is the wide area neural network which is made of the three layers which are the input layer, hidden layer and output layer [9]. The input layers consist of neurons whose building block starts from a single neuron layer model in Equation 1;

$$Y = f(wx_{ij} + b) \tag{1}$$

Where y is the output, x is the input matrix of size $(i * j)$, where $i$is the data and j is the data features; w presents the weight of the neural network, b is the bias function and f represents the activation function. Due to the diverse nature of the dataset collected with various features, three hidden layers were assumed in the modeling to improve the training computation process. These layers were formulated from the output of the neuron layer in the equation 1 which formed the input of the next three hidden layers as formulated in the Equation 2;

$$Y_L = f_l\,(w_l f_{l-1}(w_{l-1} f_{l-2}(\ldots.. f_2(w_2 f_1(w_1 x + b_1) + b_2) \ldots.. + b_{l-1}) + b_l)\,) \tag{2}$$

Where $y_l denotes$the output of the wide area neural network, b_l is the bias of the hidden layer. The number of neurons in the input layer is determined by the 22 classes of the threat data set features and also the normal packet class, while the activation function used is the hyperbolic tangent activation function. The architectural model of the neural network was presented in the Figure 1;

The Figure 1 presented the architectural model of the neural network which was remodelled with three hidden layers for training. The architectural parameters like i is the neurons, n is the number of neurons,h is the hidden layers, h_n is the number of hidden layers, o is the output layer.

2.3 Training of the Neural Network Model

Training of the neural network involves logical and arithmetic computation process which adjusted the network neurons and its properties to acquaint with the threat model features and generate a model for CARS. The steps involve the importation of the dataset and then the application of principal component analysis (PCA) [16] for the feature transformation and then feed-forward to the neural network for configuration and training using optimization algorithm.

During the training, regularization was applied to address over-fitting.The optimization algorithm used for the neural network is the gradient descent back propagation algorithm. This algorithm adjusts the hyper-parameters of the neurons during the training process while monitoring the loss function. During this process, regularization techniques are applied for generalization of weights and avoid over fitting (which is capturing noise during the training process).

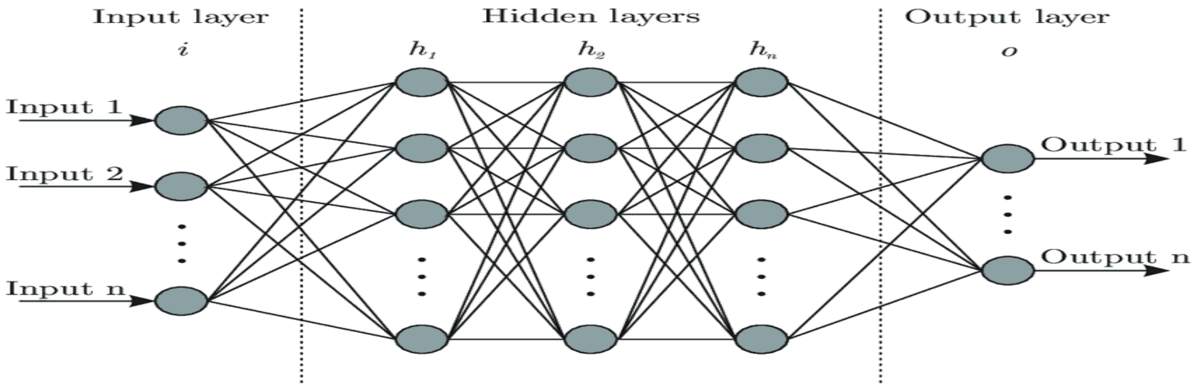

Figure 1: Architecture of the neural network with hidden layers [15]

2.4 Apply Dropout Regularization Technique during Training

Dropout regularization technique was applied for the regularization of the neural network to solve over fitting problem. The dropout operates by randomly selecting some of the neurons and turning them off through a probability vector of zero and 1 to act as the activation function and allow the generalization of the learning process to stop over fitting [12, 13]. It operates based on the principle that randomly switching on and off the neurons will give room for all the neurons to learn and generalize the loss function. The probability vector is initialized for each layer neurons using the Bernoulli $(B(p))$ function in Equation 3;

$$P_{vector}^{layer} \sim B(P) \tag{3}$$

The Equation 3 was applied to the layers to generate a probability vector of 1 and 0, and then applied to the activation unit of the network layers to generate new values of activation function as posited in the Equation 4;

$$a^{\sim layer} = P_{vector}^{layer} * a^{layer} \tag{4}$$

Where $a^{\sim layer}$ is the new activation layer for the layer and $z^{layer+1}$ is the next activation value for the next hidden layers of the network as in equation 5, while $tanh$ activation in the equation 3.6 was used to enhance nonlinearity.

$$z^{layer+1} = \left[\left(\left(a^{\sim layer}\right).T\right).dot\ \left(\theta^{layer+1}\right)\right] + b^{layer+1} \tag{5}$$

$$a^{layer+1} = tanh\ \left(a^{layer+1}\right) \tag{6}$$

Equations 4 to 6 were used to randomly dropout neurons through the assignment of zero probability which switches them off during the training process. The pseudocode of the algorithm is presented as [17];

Algorithm1: Standard Dropout algorithm:

1. Start
2. Input activation values
3. Parameter initialization (weight, bias, drop rate p)
4. Generate probability vector (p) for neurons layers with equation 4
5. Choose probability vector as (0 and 1)
6. Apply dropout of neurons using probability vector of neurons set to 0
7. Forward propagation with dropout
8. Compute the next layer activated value
9. Compute weight and bias sum of dropout
10. Apply nonlinearity with $tanh$ activation function
11. Repeat for each layer
12. Update neural network training
13. End

2.5 The Novel Adaptive Dropout Algorithm (NADA) Used

The NADA is tailored towards a dynamic control dropout process to improve training performance, reduce information loss, improve convergence time, while achieving generalization. The NADA initialized separate values for dropout probability (D_{rt}), and apply them to adjust the dropout based training progress considering loss function value of the validated data (V_l). To this end, the values used for the drop are 0.2 and 0.5 [18]. The reason for the two dropout factor was to provide adaptivity in the drop rate of neurons, and ensuring better training performance. While monitoring the training progress, increase in loss function implied degrades in the learning process while the decrease in the loss function implied improvement in the learning process. NADA due to its ability to dynamically adjust dropouts is better than the standard dropout algorithm in retaining vital unit function (information), faster convergence and overall generalizability. The NADA flow chart is presented in Figure 2 while the pseudocode is presented as:

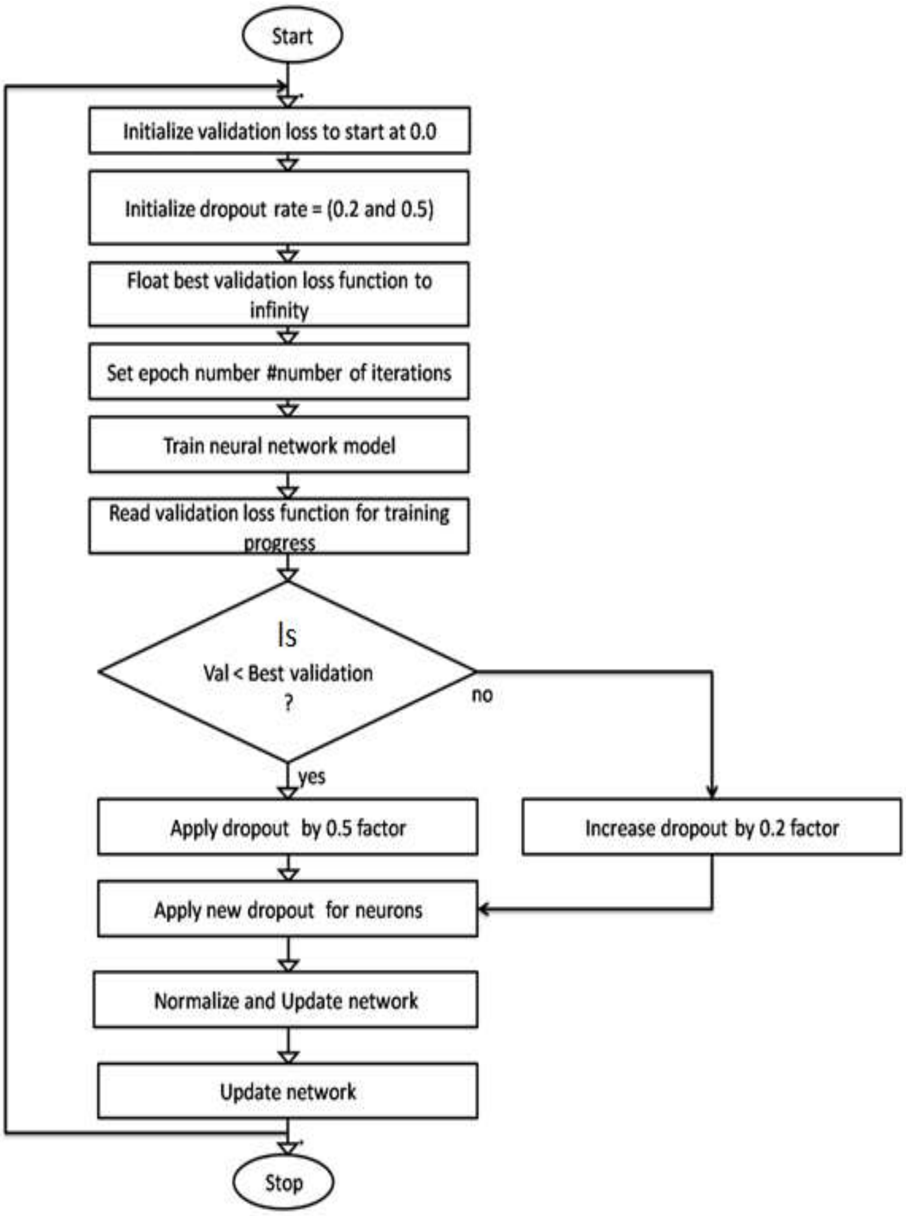
Start
Initialize validation loss to start at 0.0
Initialize dropout rate = (0.2 and 0.5)
Float best validation loss function to infinity
Set epoch number #number of iterations
Train neural network model
Read validation loss function for training progress
Is Val < Best validation ?
no
yes
Apply dropout by 0.5 factor
Increase dropout by 0.2 factor
Apply new dropout for neurons
Normalize and Update network
Update network
Stop

Figure 2: Flowchart of NADA

Algorithm 2: NADA for Improve regularization

```
Start
Initialize hyper-parameters settings
$D_{rt}$= 0.2 and 0.5 % Initial dropout probability
$V_l$ = 0.0  # Starts validation loss at 0.0
Float (Int) = $best_{V_l}$ # Initializing best validation performance for loss function as infinity
For each epoch (Number of iterations) train the model
For epoch in range (Num_epochs):
$Train_{model}$() # training of neural network
```

```
Read $V_l$ = $validate_{model}()$ # Get validation loss
Check if validation loss is less than $best_{V_l}$
If
    $V_l < best_{V_l} - I_{th}$: where $I_{th}$ is improved threshold
best$_{V_l}$ = $V_l$
    $D_{rt}$ = 0.5  # Reduce dropout rate by a factor of 0.5
Check if validation loss is greater than $best_{V_l}$
Else if
    $V_l > best_{V_l} - I_{th}$:
    $best_{V_l} = V_l$
    $D_{rt}$ = 0.2  # Increase dropout rate by a factor of 0.2
Else:
    $D_{rt}$ = 1.0  # Normalize dropouts
Apply the new dropout rate in the model
            Apply ($D_{rt}$)
End if
Return
End
```

The Figure 2 presents the NADA which was developed to improve generalization of neurons by adaptively controlling the rate of dropouts. Initially the hyper-parameter and drop rates are initialized and the drop rate set to 0.2 and 0.5. As the training of the neurons progresses, the validation loss was monitored and the values compared with the best previous loss function to determine the training progress. When the loss function decreases more than the previous values, then the training is progressing which prompts the application of more neurons to be dropped using the drop rate factor of 0.5, consequently when the training progress degrades, the drop rate is reduced through the application of 0.2 drop rate factor. These dynamic applications of drop rates considering the loss functions were used to control the learning process ensure fast convergence, and overall generalization.

2.6 Intelligent Cyber Threat Classification (ICTC) Model Generated

The intelligent cyber threat classification model was generated after the neuron converges and then loss function was tolerable according to the back-propagation algorithm. The flow chart for the generation of the classification model for the detection of the cyber threats was presented as figure 3 while the process flow of the ICTC model was presented in figure 4

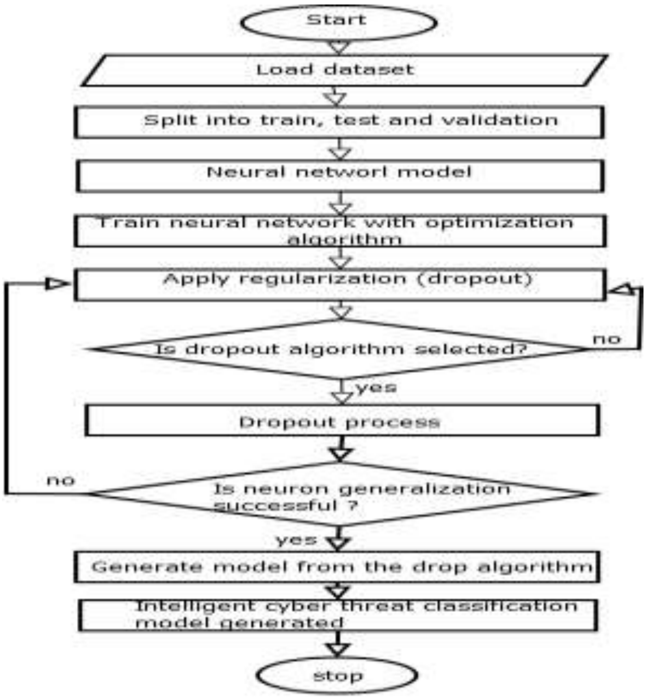
Start
Load dataset
Split into train, test and validation
Neural networl model
Train neural network with optimization algorithm
Apply regularization (dropout)
Is dropout algorithm selected?
no
yes
Dropout process
no
Is neuron generalization successful ?
yes
Generate model from the drop algorithm
Intelligent cyber threat classification model generated
stop

Figure 3: The ICTC model generation

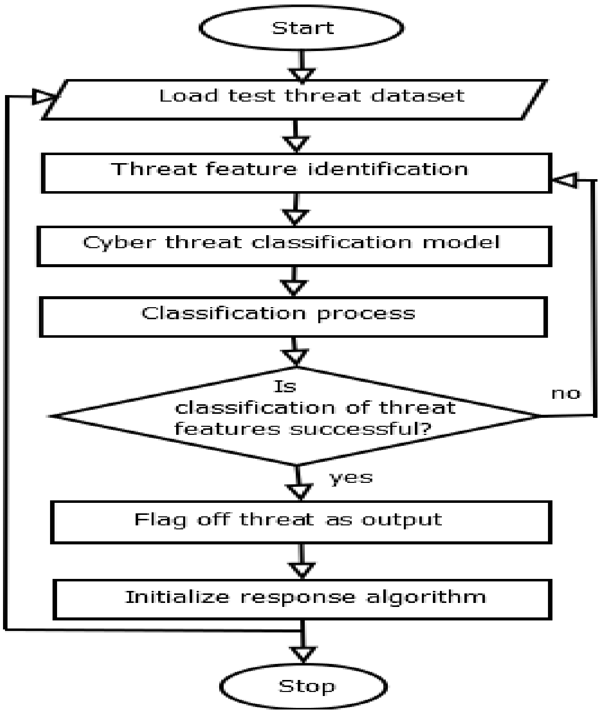
Start
Load test threat dataset
Threat feature identification
Cyber threat classification model
Classification process
Is classification of threat features successful?
no
yes
Flag off threat as output
Initialize response algorithm
Stop

Figure 4: The ICTC model operation

The Figure 3 presented how the neural network algorithm was trained with the dropout regularization technique respectively to generate the intelligence model for cyber threat detection. The model was tested using threat features as depicted in the figure 4 which showed how threat features target towards the network was identified by the ICTC model and then classified to detect the threat and then initiate the response algorithm.

2.7 Develop Cyber Threat Response Algorithm (CTRA)

CTRA is an algorithm tailored towards the isolation of the detected threat from the network facility. While the cyber incidence detection system was used for the tracking and monitoring of cyber threats targeted towards the network facility, the CTRA was used to isolate the threat to prevent harm on the network. To achieve this, data isolation algorithm was utilized to isolate the detected threats from the network. The algorithm combined access control and segmentation approach to segment the threats from the network. The segmentation focused on analysing the network information flagged from the ICTC and then applied the access control protocol to deny it access to the server, thereby isolating it from the network. The algorithm is presented as;

Algorithm 3: Cyber Threat Response Algorithm (CTRA)

```
1.  Start
2.  Initialization of ICTC model # Intelligent Cyber Threat Classification model
3.  Initialize access control protocols # MAC address filtering and rule based access protocol
4.  For
5.  Input ← $classificationmodel$ =true
6.  Then
7.  Identify threat information source
8.  Segment network information
9.  Apply access control protocols
10. Deny access to threat network protocol
11. Restrict access to identified threats
12. Login report
13. Return to step 5
14. End
```

2.8 The modelled CyberAttack Response System

The machine learning based CARS was developed using the ICTC and the threat isolation algorithm. The Figure 5 presented the process of operation of the machine learning based CARS.

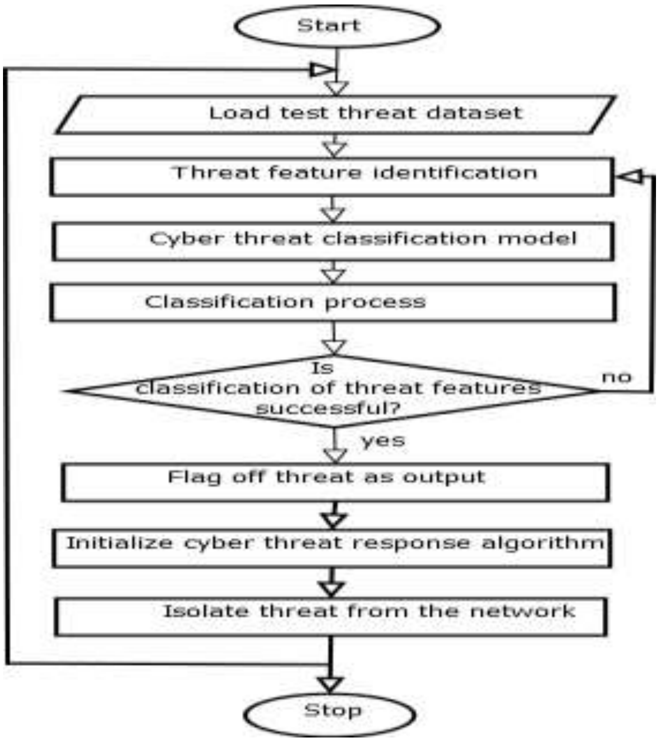
Start
Load test threat dataset
Threat feature identification
Cyber threat classification model
Classification process
Is classification of threat features successful?
no
yes
Flag off threat as output
Initialize cyber threat response algorithm
Isolate threat from the network
Stop

Figure 5: Machine learning based CARS flow chart

The Figure 5 presented the process flowchart of the CARS showing how it was used to address the issues of cyber threat impacts on 5G network infrastructure. When the testing threat data was loaded to the network, the features of the threats were identified by the ICTC model and then classified as threat. This was achieved using the neural network algorithm trained with multi classification dataset of threats and the diverse regularization algorithm which ensure that the issues of over-fitting which has the potential to affects classification model performance was addressed. The Cyber Threat Response Algorithm (CTRA) on the other hand was used to control the threat through network isolation by applying access deny protocol. Overall, the model developed was utilized to detect threats and isolate its penetration in 5G network.

2.9 System Implementation

The CARS model was developed and implemented using MATLAB. The CARS model was further segmented into the CTRA and ICTC algorithms. The ICTC, developed with a neural network algorithm, was implemented using the Classification Learner application software in MATLAB. Implementing the CTRA which is the threat isolation algorithm involves a systematic approach to enhance network security. In the initial phase, the algorithm requires initialization and setup, including the import of relevant MATLAB toolboxes, particularly the Statistics and Machine Learning Toolbox, as well as the initialization of custom models and data structures designed for threat isolation. The core of the algorithm focuses on data segmentation to divide the data into manageable units using MATLAB's data manipulation functions. Access control protocols such as Media Access Control protocol address filtering and rule-based access protocol are established to govern resource access, allowing or restricting access based on data feature behavior classified by the ICTC model. The identified threats are then evaluated in 5G networks to determine if a data block indeed poses a threat, based on predefined rules or thresholds. If a threat is confirmed, actions for isolation and denial of access are initiated, involving the execution of custom scripts or functions. Additionally, logging and reporting mechanisms are integrated into MATLAB to maintain records of actions, facilitating post-incident analysis. Continuous monitoring, testing, and validation of the algorithm ensure ongoing network security, with documentation, compliance, and training as integral components for maintaining the system's effectiveness.

3. RESULTS AND DISCUSSIONS

The neural network training reported in this section, considered the regularization algorithm proposed respectively for the generation of the intelligent cyber threats classification (ICTC) model.

The ICTC is a model generated with neural network algorithm, trained with Novel Adaptive Dropout Algorithm (NADA) regularization technique, identified as the best to address issues of over-fitting and generalized model ICTC. The integrated system was evaluated considering the five classes of threat models and the detection result was reported in Figure 6.

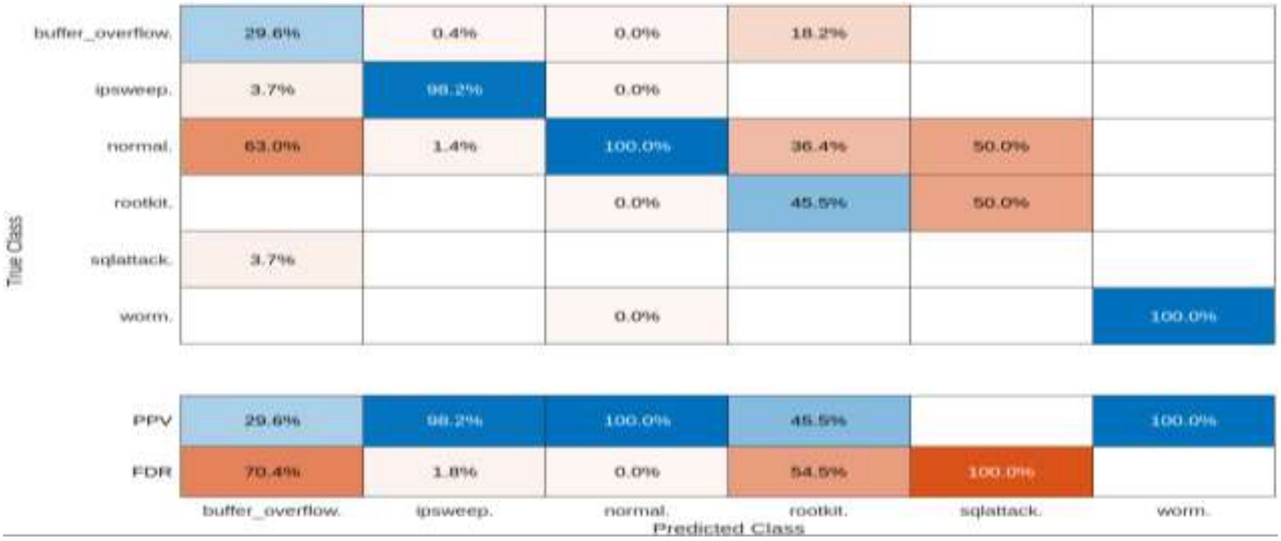

Figure 6: Confusion matrix of positive predictive value (PPV) and false detection rate (FDR)

The confusion matrix of the NADA application to neural network training for the creation of the detection model is displayed in Figure 6. The probability of correctly classifying threats was demonstrated by the positive predictive value (PPV), and the probability of incorrectly classifying threats was demonstrated by the false detection rate (FDR). Based on the results, it was noted that the rootkit and buffer overflow PPVs were less than 50%, whereas the sql attack showed no PPV. The study's imbalanced dataset was the cause of the subpar results; in contrast, the classes of normal packet, worm, and ipsweep achieved over 98% PPV classification success.

3.1 Result of the Cyber Attack Response System (CARS)

The cyber-attack response system utilized the ICTC model as input to isolate threat from the network gateway using the CTRA. First the output of the ICTC was used as input to the CTRA, which detects the malware and then isolate the threat from the network.The detection utilized the ICTC model, while the CTRA isolated it from the network environment as shown in the figure 7, displaying the average number of blocked threats.

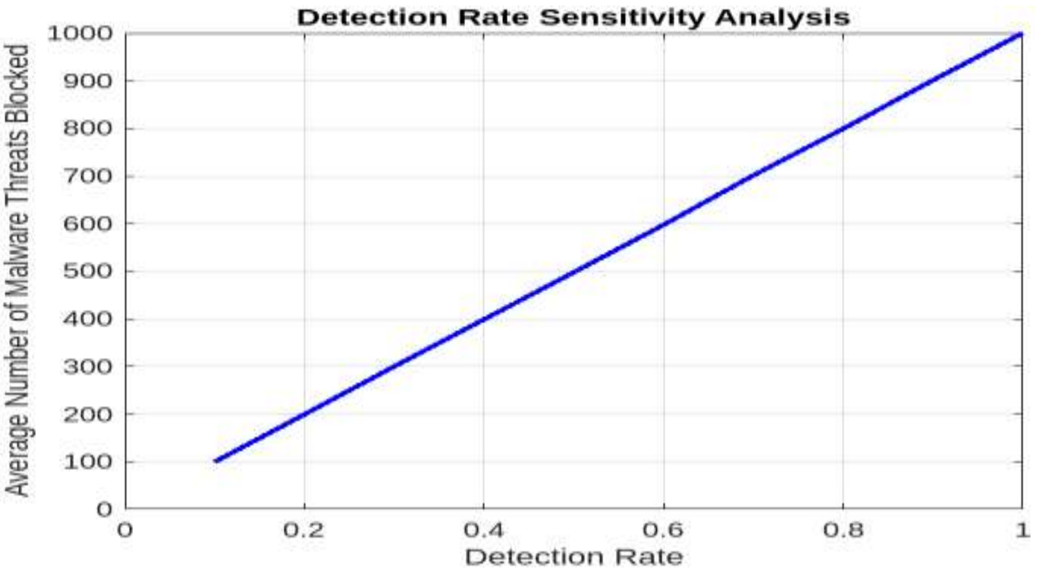

Figure 7: Result of cyber-attack response

The Figure 7 reported the result of the CARS as it detected and isolate threats from the network. The graph shows how malware are detected and blocked at every stage of its introduction to the network. The detection rate as the malware feature increases also increases. This implied that as the threats penetration volume increases, the sensitivity of the detection model also increases, and at each stage, the detected threats are blocked.

3.2 Result of Programmed CTRA Firewall against Malware attack

This section used programming to evaluate the integration of the CTRA on the Nigerian Television Authority (NTA) network infrastructure. To achieve this, data of the physical network was used to set a virtual network environment, and then malware was simulated to the network, through the CTRA based firewall and the results was evaluated considering the threat detection and blockage performance and then impact on quality of service. Table1 showed NCC or 3GPP Standard for network analysis. Table 2 showed the simulation parameters. The figure 8 showcased the result of the testing process.

Table1: Standard for LTE network analysis

Network condition	Throughput (%)	Packet loss (%)	Latency (ms)
Excellent	> 80	< 2.5	< 20
Good	70 to 79	3 to 4	20 to 50
Fair	50 to 69	4 to 5	51 to 100
Very bad	< 49	>5	> 100

Table 2: Simulation parameters

Simulation Variable	Example Values
Network Topology	Single Access Point, Mesh Network, Star Topology
Number of Access Points	1, 3, 5
Wireless Standards	802.11ac, 802.11ax (Wi-Fi 6), 5G NR-U
Frequency Bands	2.4 GHz, 5 GHz, 60 GHz
Threat Scenarios	Buffer_overflow, Ipsweep, Portsweep, Rootkit, and normal packet
Users equipments	3
Access control protocols	
Security Policies	MAC Filtering, Guest Network Isolation
Key Management	Pre-Shared Keys (PSK), 802.1X (RADIUS)
Security Updates	Firmware Updates, Patch Management
Intrusion Detection System	ICTC model
User Behavior	Normal Usage, Anomalous Behavior

Simulation Variable	Example Values
Compliance and Standards	IEEE 802.11i, WPA3, WPA2, WPA, IEEE 802.1X
Real-Time Monitoring	Yes, No
Performance Metrics	Latency, Throughput, Loss

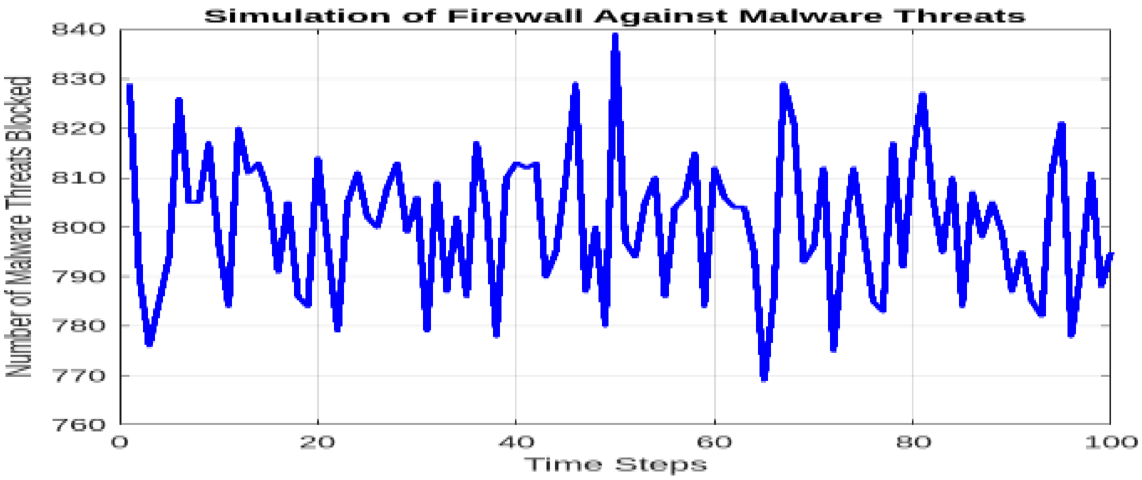

Figure 8: Threat detection performance with CTRA

The Figure 8 showcased the threat detection performance of the CTRA as it detects and isolate threat from the virtual network. To view the quality-of-service performance, which demonstrates the effectiveness of the CTRA on the network, throughput, latency and loss were measured as reported in the Figure 9, 10 and 11.

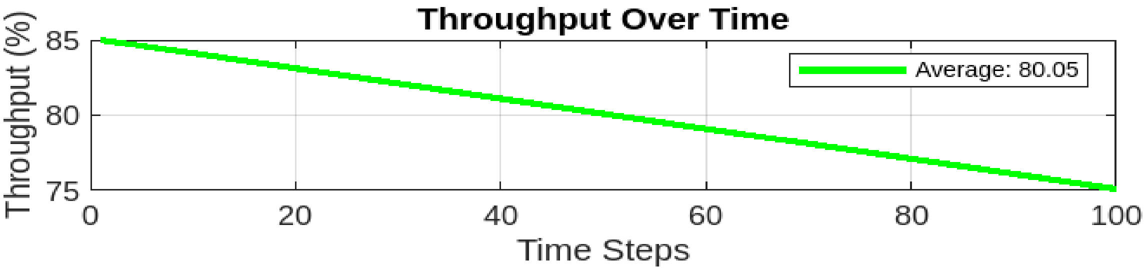

Figure 9: Throughput performance with CTRA during malware input

The Figure 9 showcased the throughput performance with CTRA. From the result it was observed that while the malware penetrated the network, the CTRA based firewall detected and isolated it from the network. The average latency of the network is 80.05%, the throughput is excellent. What this means is that the malware features targeted towards the network to cause denial of service and potential ransomware was detected and isolated from the network, which had no impact on the throughput of normal packet flow and hence maintained quality of service.

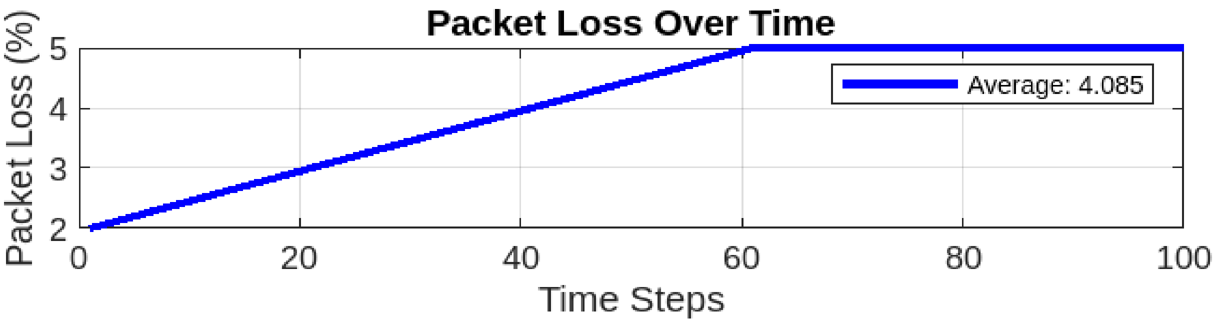

Figure 10: Loss with CTRA during malware input

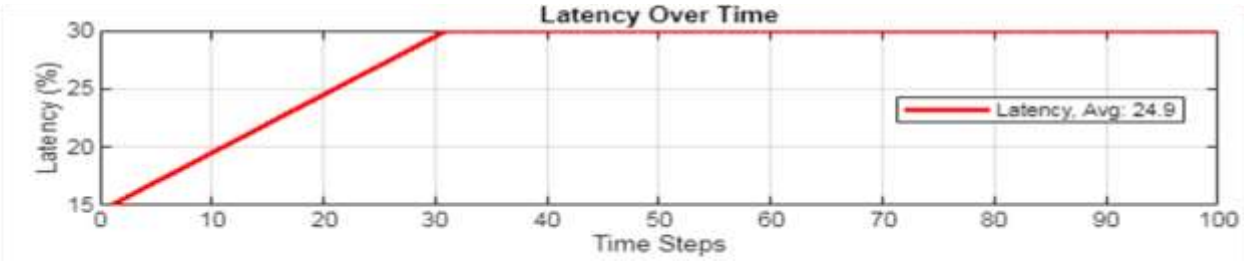

Figure 11: Latency with CTRA during malware input

The Figure 10 showcased the packet loss with CTRA, with an average loss rate of 4.085%, while latency reported in figure 11 was an average of 24.9ms. The reason the latency and the packet loss remained constant after some time was because the network router was adjusting to the network load and hence resulted the constant latency reported, which by the way is good as the average is below 30ms. Overall, the results showcased that the CTRA firewall was very effective, as it detected and isolated malware targeted at the network infrastructure and isolated it from the gateway, while allowing normal packet access to the server.

4. CONCLUSION

This paper on modelling of cyber-attack response system for 5G network using machine learning techniques aims at address the evolving landscape of cybersecurity threats in 5G networks. Through the development and implementation of innovative machine learning techniques, the project contributed significantly to enhancing the security posture of 5G networks. By introducing novel approaches such as dropout technique, and intelligent cyber threat classification models, the study has laid the groundwork for more robust and effective cyber-attack response systems in the era of 5G connectivity. The results when tested with simulation considering quality of service such as throughput, latency and packet loss reported 80.05% for throughput, 4.09% for packet loss, 24.9ms for latency, these results when compared with standards for best practices (table 1) in 5G network, demonstrated the effectiveness of the new security model developed against cyber threats. Specifically for malware, during system integration of the model on 5G network, the throughput reported 71.81%. In addition, packet loss reported loss rate of 23.18%, while latency reported 178.98ms. The findings of the study hold great promise for improving the resilience and security of 5G networks against cyber threats, thereby safeguarding critical infrastructure and ensuring the integrity of digital communications in the modern era.

REFERENCES

[1] Imanbayev, A., Tynymbayev, S., Odarchenko, R., Gnatyuk, S., Berdibayev, R., Baikenov, A.&Kaniyeva, N. (2022). Research of machine learning algorithms for the development of intrusion detection systems in 5G mobile networks and beyond. Sensors, 22, 9957. https://doi.org/10.3390/s22249957.

[2] Neha Yadav, Sagar Pande, Aditya Khamparia& Deepak Gupta (2022). Intrusion Detection System on IoT with 5G Network Using Deep Learning. Wireless Communication and Mobile Computing, ID 9304689. https://doi.org/10.1155/2022/9304689

[3] Casillas, R., Touchette, B., Tawalbeh, L. & Muheidat, F. (2020). 5G Technology Architecture: Network Implementation, Challenges and Visibility. Int. J. Comput. Sci. Inf. Secur. 18(1), 39–53.

[4] Maksim Iavich, Giorgi Iashvili, ZhadyraAvkurova, Serhii Dorozhynskyi & Andriy Fesenko (2021). Machine Learning Algorithms for 5G Networks Security and the Corresponding Testing Environment. CPITS-II-2021: Cybersecurity Providing in Information and Telecommunication Systems, 3187, 139-149.

[5] Obodoeze Fidelis C. and Francis A. Okoye (2018). Holistic Security Implementation. Journal of Trend in Scientific and Development (IJTSRD), 2(2), 598-607.

[6] Mozo, A., Pastor, A., Karamchandani, A., de la Cal, L., Rivera, D. & Moreno, J.I. (2022). Integration of Machine Learning-Based Attack Detectors into Defensive Exercises of a 5G Cyber Range. Appl. Sci. 2022, 12, 10349. https://doi.org/10.3390/app12201034.

[7] Sultana, N., Chilamkurti, N., Peng, W. & Alhadad, R. (2019). Survey on SDN Based Network Intrusion Detection System Using Machine Learning Approaches. *Peer-Peer Netw. Appl.* **2019(***12)*, 493–501.

[8] Ogbeta L.K. & Nwobodo Lois (2023). Neuro based strategy for real time protection of wireless network ecosystem against DDOS attack. [J] I1SRED, 5(3), 79-98.

[9] Ogbuanya I.M. & Eke James (2023).Detection And Isolation Of Black-Hole In Wireless Broadband Ecosystem Using Artificial Intelligence. International Journal Of Real Time Applications And Computing System (IJORTACS), 2(2), 390-402.

[10] Oduah O. & Olofin B.B., (2023). Development of Multi Level Intrusion Detection System For Cloud Based Log Management Using Machine Learning Technique. International Journal of Real Time Applications And Computing System (IJORTACS), 1(7), 101-114.

[11] Beck, K., Beedle M. & Grenning J. (2001). The Agile Manifesto. Agile Alliance.http://agilemanifesto.org/

[12] Sun W., Tang J.&Bai C. (2019). Evaluation of university project based on partial least squares and dynamic back propagation neural network group.*IEEE Access*, 7, 69494–69503.

[13] Feng W., J. Tang &Liu T. X. (2019). Understanding dropouts in moocs,*AAAI*, 33, 517–524.

[14]The IEEE website [online]. Available: https://ieee-dataport.org/documents/5g-nidd-comprehensive-network-intrusion-detection-dataset-generated-over-5G-wireless

[15] Birhakahwa, Kelvin &Tartibu, Lagouge. (2023). Enhancing Grain Moisture Prediction with Artificial Neural Networks and Computational Fluid Dynamic. International Conference on Artificial Intelligence and its Applications 2023. 181-188. https//doi.org/10.59200/ICARTI.2023.026

[16] Pechenizkiy, M., Tsymbal A. &Puuronen S., (2004). PCA-based feature transformation for classification: issues in medical diagnostics. Proceedings. 17th IEEE Symposium on Computer-Based Medical Systems, Bethesda, MD, USA, 535-540, doi: 10.1109/CBMS.2004.1311770.

[17] Salehin, I., & Kang, D. K. (2023). A Review on Dropout Regularization Approaches for Deep Neural Networks. Electronics, 12(14), 3106. doi: 10.3390/electronics12143106

[18] Brownlee, J. (2019). A Gentle Introduction to Dropout for Regularizing Deep Neural Networks. Machine Learning Mastery. https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/