Exploromatics
Journal of Innovative
Engineering and Technology
Volume: 05 No: 02 | August -2024
ISSN (Online) 2636 – 590
ISSN (Print) 2636 - 591X

CYBER-THREAT DETECTION MODEL USING ARTIFICIAL NEURAL NETWORK AND NOVEL ADAPTIVE DROPOUT ALGORITHM FOR 5G NETWORK

Kwubeghari Anthony[*1], Chibueze Kingsley I. [2], Okoye Francis A [1]
1 Department of Computer Engineering, Enugu State University of Science and Technology
2 2Department of Computer Science & Maths, Godfrey Okoye University, Enugu
Author for correspondence: Kwubeghari Anthony; Email: kwubeghari@gmail.com

Abstract - This paper addresses the optimization problem during the training of a cyber-threat detection system with a neural network using a Novel Adaptive Dropout Algorithm (NADA). Data was collected from the Institute of Electrical and Electronics Engineering (IEEE) Dataport, which is an open repository for studies. The sample size of the data collected is 125871 samples, consisting of 41 features of threats across 22 threat classes. Principal Component Analysis (PCA) is the feature transformation technique utilised for data processing. The neural network model utilised for the study is the wide-area neural network, which is made of three layers: the input layer, the hidden layer, and the output layer. 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. Regularisation techniques were used in the training process to address the issues of overfitting of neurons and generalisation of weights. The study then adopts a new dropout algorithm that is tailored towards a dynamic control dropout process to improve training performance, reduce information loss, improve convergence time, and achieve generalization. The result of the proposed technique is a high performing approach, as it achieved an average Area Under Curve (AUC) of 0.9383 on average.

Keywords: Cyber Threat; Artificial Neural Network; NADA; PCA; Back Propagation

1. Introduction

The architecture of 5G is designed with highly advanced network elements and terminals to enable a new scenario (Arabo and Pranggono, 2013). Additionally, service providers can easily adopt advanced technology to offer value-added services. The system is based on an all-Internet Protocol (IP) model for interoperability between wireless and mobile networks (Mantas, et al., 2015). The technology used in 5G is called 5G New Radio Technology, and it is based on Orthogonal Frequency Division Multiplexing (OFDM) (Shariat, et al., 2019), which is a way of modulating digital signals across several different channels to reduce interference. In addition to this, 5G also uses wider bandwidth technologies, such as 6 GHz and mm wave (Cao, et al., 2019). One of the aims of 5G is to reach a maximum data rate of 10 GB, which is ten times faster than the maximum data rate of 4G, which is currently at 1 GB (Shariat, et al., 2019). Therefore, 5G is designed to achieve higher data rates, more capacity, and less delay than current 4G radio access technologies (LTE/LTE Advanced) by using a 5G radio access technology (Tian, et al., 2019).

The security challenges associated with 5G technology are significant and include the need to ensure the safety of critical network infrastructure and user privacy in an environment where all devices are connected to the internet and exposed to a variety of potential attacks (Salahdine, 2018; Lai, et al., 2020). For example, if there is a security breach in a smart grid system, it could lead to damage to the electrical system and harm other interconnected systems and services. Additionally, user privacy is at risk when transmitting sensitive data over the 5G network. Therefore, there is an urgent need to develop security solutions that can protect the

5G network while ensuring high data rates and low latency. The classification of security challenges and issues is dependent on the specific 5G use case involved (Liu et al., 2017; Tran et al., 2017).

There are four main classes of security attackers in 5G: insider, outsider, network, and virus. An insider attacker is someone who tries to impact the control and execution functions of a system to change their behaviour. An outsider attacker aims to influence the communication system by either monitoring data or gaining access to sensitive data (Mijumbi, et al., 2016). Network attackers attempt to shut down or disrupt the functioning of a network, while virus attacks use software to gain access to a system for malicious purposes. These attacks can be classified into two categories: attacks targeting the user and attacks targeting the network. Examples of attacks that fall under the user category include device triggers, node capture, and privacy leaks (Yang and Fung, 2016; Ahmad, et al., 2018). Machine learning algorithms can be used to analyse network traffic and identify patterns of behaviour that may indicate a security threat. Machine learning can also be used to identify anomalies in network traffic that may indicate an attack (Salahdine and Kaabouch, 2019). Machine learning (ML) can play a vital role in threat detection and response for 5G networks. With the increasing complexity of 5G networks, ML can be used to analyse large volumes of network data and identify anomalous behaviour that may indicate a security threat (Neha, et al., 2022).

There are several research studies that focus on different aspects of security in 5G networks, such as (Tata and Kadoch, 2022; Bocu and Iayich, 2023; Tomida, et al., 2021; Maksim, et al., 2021). Some of the studies focus on specific security threats, such as IP spoofing and energy-efficient security, while others focus on specific techniques, such as network coding and client-server key management. Among the studies, Maksim, et al., (2021) considered 15 different types of attacks with the potential to penetrate 5G networks without obtaining a solution to protect the 5G network against the threats, and this has remained a gap. This research proposes to address this gap through the adoption of a novel adaptive dropout algorithm for cyber-attack detection in the network using an artificial neural network.

2. Data Collection

This study characterized the 5G network facility at the ICT Department, Nigerian Television Authority, Headquarters, Abuja, which is the primary source of data collection. The main ICT component considered for the characterization is the 5G NR V/ADSL2+Wifi 6 AX1500 VPN firewall system, which used the WPA-PSK security protocol for the network protection against intrusion.

Table 1: Results of Characterization

Time (min)	Data upload (Mb)	Throughput (Mbps)	Latency (ms)	Loss (%)	Throughput (%)
1	229.5385	157.5454	88.22528	10.61914	68.6357
2	252.042	172.9908	94.36878	12.85392	68.6357
3	267.7983	181.6954	105.71272	13.60553	67.84787
4	279.492	184.7795	120.89956	13.81089	66.11261
5	290.9112	191.9815	127.60416	14.1844	65.99317
6	300.7383	196.044	136.61352	15.056	65.18759
7	321.6003	208.7405	141.21906	15.74506	64.90682
8	333.0634	213.4754	149.12958	15.91183	64.09453
9	345.2567	220.3107	151.18184	16.7409	63.8107
10	355.1442	224.0808	156.02518	17.29182	63.09573
11	374.469	235.662	184.30425	17.57388	62.9323

12	378.1967	235.9634	188.36396	19.9367	62.3917
13	383.264	237.6708	196.10316	20.35577	62.0123
14	394.7875	243.5286	203.8366	24.57148	61.686
15	395.9019	242.0374	205.19078	24.7158	61.1357
16	399.6187	243.729	210.08189	25.8883	60.9904
17	400.8923	244.2108	212.08808	27.211	60.9168
18	405.1636	245.1856	214.33903	27.8211	60.5152
19	405.8333	245.0555	214.60944	31.5353	60.3833
20	410.0497	247.3211	216.1005	32.6885	60.3149
21	411.2684	247.1468	222.97236	34.5785	60.0938
22	448.9793	268.3132	230.34583	35.71915	59.7607
23	460.5577	273.4096	234.604465	39.2345	59.3649
24	471.5322	279.6983	234.606134	40.06951	59.3169
25	477.143	281.0735	234.724496	43.3994	58.9076
26	488.6006	286.3224	333.784991	44.6423	58.6005
27	498.5485	283.704	346.390967	45.88602	56.906
28	499.1633	274.9491	407.050771	49.4353	55.082
29	504.4722	277.0566	435.242097	50.08659	54.9201
30	510.6814	267.975	445.557825	53.30433	52.474
Avg.	389.8236	240.0047	214.7092	27.81576	61.56752

The table 1 presented the result of the network characterization considering the quality of service when malware was simulated for 30minutes. The result analyzed average on 389.82Mb packet data infected with malware reported an average latency of 214.71ms, loss of 27.82% and throughput of 61.57%. What this mean is that the existing network security model was not able to detect and differentiate the malware features from the packet data and this as a result impacted on the server performance, leading to poor KPI results for throughput, latency and losses, when compared with the standard for best practices.

While the primary data collection discussed earlier focused on the network characterization data reports, 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 and Electronics Engineering (IEEE) Dataport, which is an open repository for studies. The sample size of the data collected is 125871 samples, consisting of 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. Table 2 presents the characterization of the data features

Table 2: Data Description of threat features

Feature Name	Data Type	Description
Duration	Integer	time used for the connection
Protocol Type	Categorical	The network protocol types
Services	Categorical	The service request type provided
Flag	Categorical	The associated flags with the connection

Src_byte	Integer	The byte size of packet sent from source to destination
Dest_byte	Integer	Size of byte transferred from destination to source
Land	Binary	This indicated the connection source e.g host
Wrong_fragment	Integer	Wrong fragments number received
Urgent	Integer	Number of urgent packets
Hot	Numeric	Level of hotness of the connection
Num_failed_logins	Numeric	failed login attempts rate
Logged_In	Binary	If user is successfully logged in
Num_compromised	Numeric	Number of conditions compromised
Root_shell	Binary	Determines if root shell is obtained
Num_root	Integer	Number of accessed root
Num_file_creation	Integer	Number of files created
Num_shells	Integer	Number of prompted shells
Num_access_files	Integer	Number of files access
Num_outbound_cmds	Integer	Number of commands that is outbound
Is_host_login	Binary	Indicates host login
Is_guest_login	Binary	Indicates guest login
Count	Integer	Number of same host connection
Srvr_count	Integer	Number of same service connection
Serror_rate	Integer	Error rate for connections
Rerror_rate	Integer	Error rate for receiver side
Srvr_reror_rate	Integer	Error rate for connections to the same service
Same_srvr_rate	Integer	Rate of connections to the same service
Diff_srvr_rate	Integer	Rate of connections to different services
Same_serve_rate	Integer	Rate of connections to the same server
Srvr_diff_host_rate	Integer	Rate of connections to different hosts on the same service
Dest_host_count	Integer	Number of connections to the same destination host
Dest_host_same_srvr_rate	Integer	Rate of connections to the same service on the destination host
Dest_host_diff_srvr_rate	Integer	Rate of connections to different services on the destination host
Dest_host_same_src_port_rate	Integer	Same source port connection to destination host
Dest_host_srvr_diff_host_rate	Integer	Different hosts to destination server connection
Dest_host_serror_rate	Integer	Error rate for connections to the destination host
Dest_host_srvr_serror_rate	Integer	Error rate for connections to the destination server
Dest_host_rerror_rate	Integer	Error rate for connections to the destination host
Dest_host_srvr_rerror_rate	Integer	Error rate for connections to the receiver side

2.1 Feature Transformation

Principal Component Analysis (PCA) is the feature transformation technique utilised for data processing (Pechenizkiy, et al., 2004). It operates by identifying the most significant patterns in the data, known as principal components, and projecting them onto a lower-dimensional space while preserving as much variance as possible through the determination of orthogonal linear combinations of the original features that maximise the variance. This way, the collected data was transformed into an identifiable feature vector for training.

3. Artificial Neural Network Modelling

The neural network model utilised for the study is the wide area neural network, which is made of three layers: the input layer, hidden layer, and output layer (Ogbuanya and Eke, 2023). The input layers consist of neurons, whose building blocks start from a single neuron layer model in equation 1;

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

Where Y is the output, X is the input matrix of size where is the data and is the data features; W presents the weight of the neural network, b is the bias function, and represents the activation function. Due to the diverse nature of the dataset collected with various features, three hidden layers were assumed in the modelling to improve the training computation process. These layers were formulated from the output of the neuron layer in equation 1, which formed the input of the next three hidden layers as formulated in equation 2.

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

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;

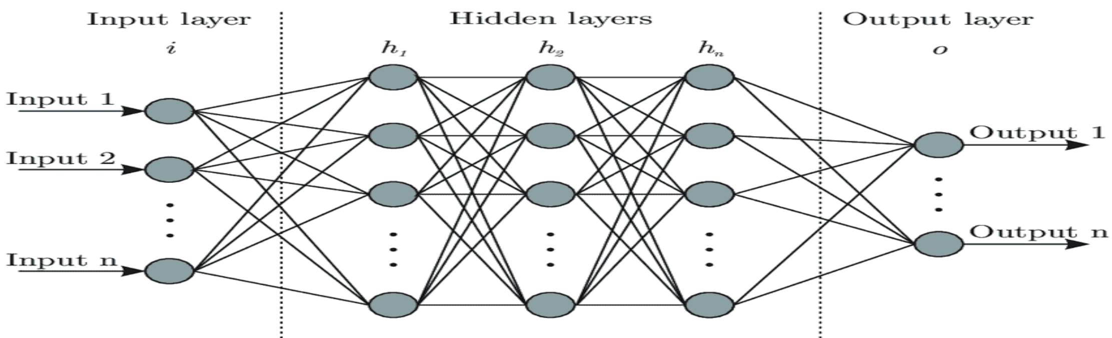

Figure 1: Architecture of the neural network with hidden layers (Birhakahwa and Tartibu, 2023)

Figure 1 presented the architectural model of the neural network, which was remodelled with three hidden layers for training. The architectural parameters are: the neurons, the number of neurons, the hidden layers, the number of hidden layers, and the output layer.

3.1 Training of the Neural Network Model

Training of the neural network involves a logical and arithmetic computation process that adjusts the network neurons and their properties to acquaint themselves with the threat model features and generate a model. As shown in Figure 2, the steps involve the importation of the dataset, then the application of Principal Component Analysis (PCA) for the feature transformation, and then feed-forward to the neural network for configuration and training using an optimisation algorithm. During the training, regularisation was applied to address overfitting.

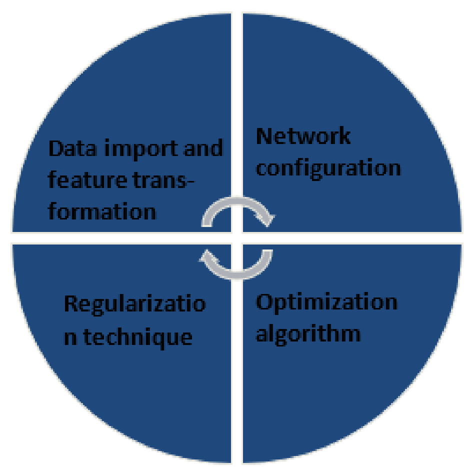

Figure 2: Neural Network Training Lifecycle

3.2 Training Optimization algorithm

The optimisation 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, regularisation techniques are applied to generalise weights and avoid overfitting, which captures noise during the training process (Ogbeta and Nwobodo, 2022).

3.3 Diverse Regularization Algorithms for the Generalization of Neurons

Regularisation techniques were used in the training process to address the issues of overfitting of neurons and generalisation of weights. In the regularisation process, three techniques were presented, respectively: the Novel Adaptive Dropout Algorithm (NADA), the Assembled Regularised Approach (ARA), and the Standard Dropout Approach.

4. The Novel Adaptive Dropout Algorithm (NADA)

The new dropout algorithm is tailored towards a dynamic control dropout process to improve training performance, reduce information loss, improve convergence time, and achieve generalization. The NADA initialised separate values for dropout probability (D_{rt}), and applied them to adjust the dropout-based training progress considering the loss function value of the validated data (V_l). During the training process, the loss function is monitored depending on its increases or decreases in behaviour, and then the output is used to inform the application of the dropout factor for randomly selected neurons. According to (Srivastava, et al., 2014; Zhang, et al., 2017; Brownlee, 2019), dropout values of 0.2 to 0.5 are good for assignment, because high dropout factors like 0.9 delay convergence and may not allow the neurons to learn properly. To this end, the values used for the drop are 0.2 and 0.5. The reason for the two-dropout factor was to provide adaptivity in the drop rate of neurons, and ensure better training performance. While monitoring the training progress, an increase in the loss function implied degradation in the learning process, while a 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. In addition, the adaptation of the dropout rate ensures equilibrium between learning and regularisation to improve the overall learning process. The NADA pseudocode is presented as;

Algorithm 1: NADA for Improve regularization

1. Start
2. **Initialize hyper-parameters settings**
3. D_{rt}= 0.2 and 0.5 % Initial dropout probability
4. V_l = 0.0 # Starts validation loss at 0.0
5. $Float\ (Int)\ =\ best_{V_l}$ # Initializing best validation performance for loss function as infinity
6. **For each epoch (Number of iterations) train the model**
7. For epoch in range (Num_epochs):
8. $Train_{model}$() # training of neural network
9. $Read\ V_l$ = $validate_{model}$() # Get validation loss
10. **Check if validation loss is less than** $best_{V_l}$
11. If
12. V_l<$best_{V_l}$- I_{th}: where I_{th} is improved threshold
13. $best_{V_l}$ = V_l
14. D_{rt} = 0.5 # Reduce dropout rate by a factor of 0.5
15. **Check if validation loss is greater than** $best_{V_l}$
16. Else if
17. V_l>$best_{V_l}$- I_{th}:
18. $best_{V_l}$= V_l
19. D_{rt} = 0.2 # Increase dropout rate by a factor of 0.2
20. Else:
21. D_{rt} = 1.0 # Normalize dropouts
22. **Apply the new dropout rate in the model**
    Apply (D_{rt})
23. End if
24. Return
25. End

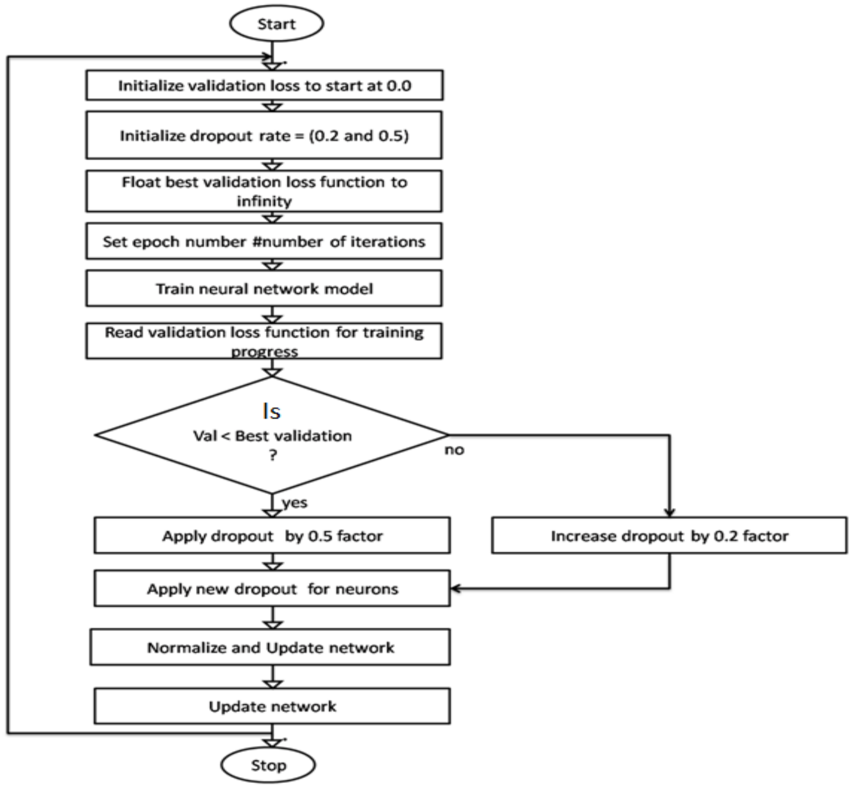

Figure 3: Flowchart of NADA for Improved Regularization

5. Result Of ANN with NADA

In the NADA regularisation technique, the dropouts of 0.2 and 0.5 were adaptively applied based on the gradient loss output during the training process. This gradient loss was used to evaluate the neuron learning rate, and then, based on the outcome, the appropriate dropout factor was applied to the neurons to allow for a more generalised model. The training of the neural network utilised this NADA regularisation technique and back-propagation algorithm, to adaptively adjust the neurons to learn threat features and generate the detection model. The results from the training process, which were generated in a MATLAB environment, were reported as figures 4 to 7.

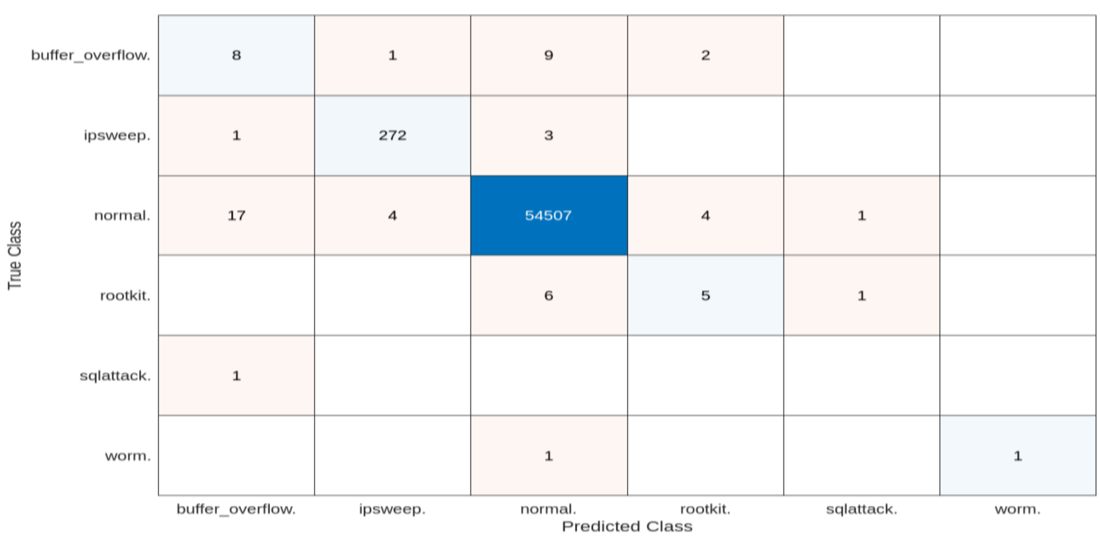

Figure 4: Confusion matrix result with NADA

Figure 4 showcases the result of the neural network training with NADA. The result showed the distribution of malware features across the six classes of the dataset. From the result, it was observed that the normal packet occupies the largest portion of the data with 54533 samples. This result confusion matrix feature distribution demonstrated the imbalance nature of the dataset, which hence made it the perfect data model for the evaluation of the regularisation models, respectively. To measure the rate of correct classification and false classification, figure 5 was applied.

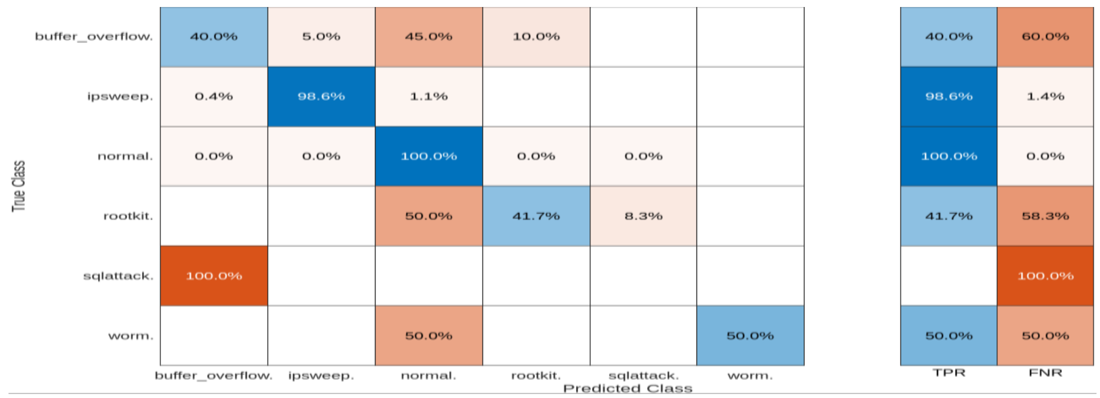

Figure 5: Confusion matrix of True Positive Rate (TPR) and False Negative Rate (FNR) with NADA

Figure 5 showcases the results of true positive classification and false negative classification, respectively, for each class of the dataset. From the result, it was observed that buffer_overflow threats recorded a True Positive Rate (TPR) of 40% and False Negative Rate (FNR) of 60%, while rootkit threats recorded 41.7% TPR and 58.3% FNR. Overall, this set of results showcased the poor classification performance of the model for rootkit and buffer–overflow threats. However, the results of normal packet and ipsweep classification, showcased a correct high TPR of 98.6% and a 1.4% FNR. Normal packet classification reported 100% TPR and 0% FNR. In addition, worm classification reported 50% TPR and 50% FNR. Overall, it was deduced from the results that the model was good, but was not able to effectively classify certain structured query language attacks (sql attacks) due to the class imbalance of the dataset; while those classes that recorded poor TPR were also due to a deficiency of the class (class imbalance) in the dataset. However, the classes of normal packet, and ip-sweep recorded a good classification success rate. In the next results, the Positive Predictive Value (PPV) and False Detection Rate (FDR) were examined, respectively, with the NADA regularisation algorithm

.

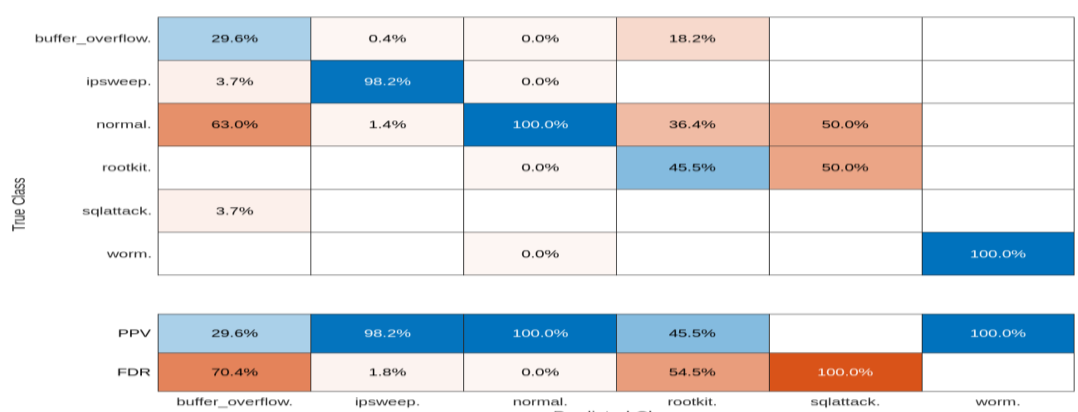

Figure 6: Confusion matrix of PPV and FDR with NADA

Figure 6 showcases the confusion matrix of the NADA application to neural network training for the generation of the detection model. The positive predictive value (PPV) showcased the probability of correct classification of threats, while the false detection rate (FDR) showed the probability of incorrect threat classification. From the result, it was observed that the PPV for buffer overflow and rootkit was recorded below 50%, while the sql attack reported no PPV. The reason for the poor results was due to the imbalance nature of the dataset used for the study; meanwhile, the classes of normal packet, worm, and ipsweep recorded over 98% PPV classification success. To measure the relationship between true positive and false positive, the area under the curve was applied to the six classes of threat features as depicted in figure 7;

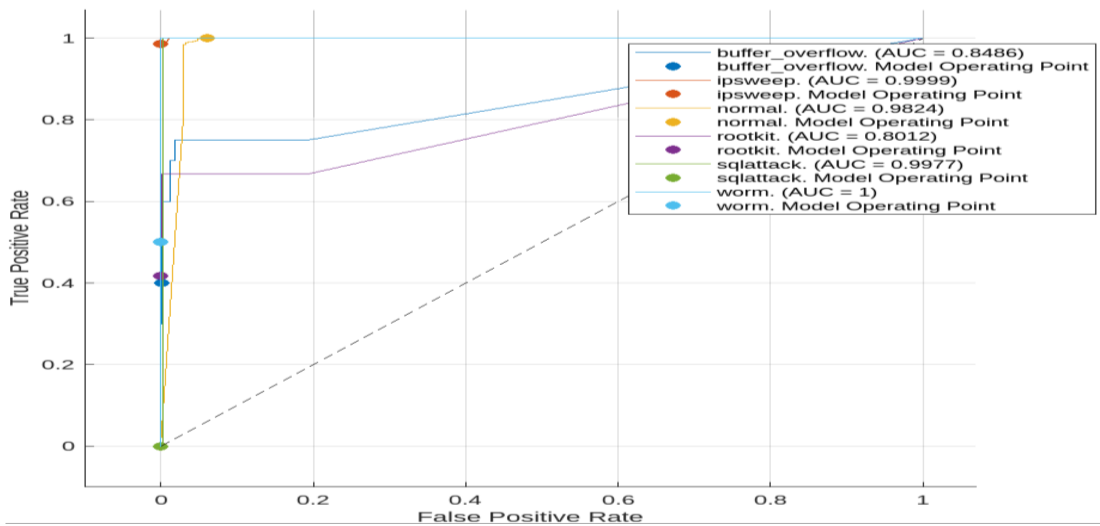

Figure 7: Area Under Curve (AUC) with NADA

The AUC is a tool used to demonstrate the relationship between TPR and FPR for each of the malware classes and normal packets. The aim of the AUC is to record a value equal to or approximately 1, which thus indicates the ability of the model to correctly classify threats and also correctly classify normal packets, respectively. From the graphs, it was observed that buffer overflow is 0.8486, ipsweep is 0.9999, normal packet is 0.9824, rootkit is 0.8012, sql attack is 0.9977, and worm reported 1.00. Overall, from the results of the AUC, it can be detected that the model with NADA was able to correctly classify normal and malware, however, referring to the confusion matrix result, it can be deduced that the AUC, despite the effectiveness of the tool, does not completely define the success of classification models, because some of the models that recorded high AUC, such as the SQL attack and buffer overflow, for instance, actually reported poor performance in the confusion matrix.

5.1 Comparative analysis of regularization techniques

The comparative analysis was applied to identify the best model for the development of the cyber threat detection system. The comparative analysis considered the machine learning models, their AUC performance in the detection of cyber threats, as depicted in Table 3.

Table 3: Comparative AUC performance of different techniques

Author (Year)	Technique	Performance AUC
Bebeshko et al., (2021)	K-Nearest Neighbour (KNN)	0.9133
Azeez et al., (2023)	Random Forest	0.9582
Dhanya et al., (2023)	Deep Multi-Layer Perceptron + ADAM optimizer	0.9790

Alshehri et al., (2022)	RNN-LSTM	0.9700
Kravichik and Shabtai (2018)	CNN + LSTM	0.9670
Huang and Chang (2019)	Random Forest	0.9300
Proposed model	ANN+NADA	0.9383

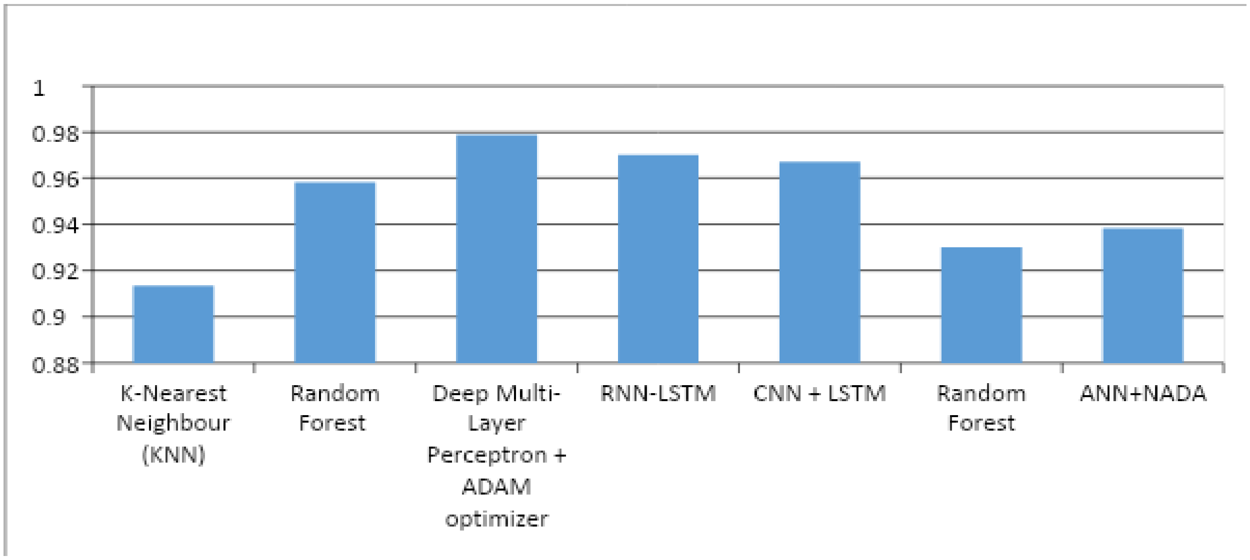

Figure 8: Comparative Analysis of Results

Table 3 and Figure 8 show that the proposed model (ANN+NADA) is not the top performing technique, as Dhanya et al., (2023) with the Deep MLP technique achieved an average AUC of 0.9790, followed by Alshehri et al., (2022) with (RNN+LSTM) technique, which achieved the second highest AUC with a result of 0.9700. However, our proposed technique is a high performing approach, as it achieved an average AUC of 0.9383. This result could turn out to be the best result, as the dataset used for training and testing the model has the highest rate of 41 features, which were not considered by the other works. Secondly, the data used in this work equally considered a higher number of cyber-attack types, where buffer overflow is 0.8486, ipsweep is 0.9999, normal packet is 0.9824, rootkit is 0.8012, sql attack is 0.9977, and worm reported 1.00 in AUC performances. These types of attacks were not considered in the other works.

6. Conclusion

This study proposes to address this problem by adopting the Novel Adaptive Dropout Algorithm (NADA) for cyber-attack detection in the network using an artificial neural network. Data was collected from the Institute of Electrical and Electronics Engineering (IEEE) Dataport, which is an open repository for studies. The neural network model utilised for the study is the wide area neural network, which is made of three layers: the input layer, the hidden layer, and the output layer. The optimisation 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. Regularisation techniques were used in the training process to address the issues of overfitting of neurons and generalisation of weights. The study then adopts a new dropout algorithm that is tailored towards a dynamic control dropout process to improve training performance, reduce information loss, improve convergence time, and achieve generalization. The result of the proposed technique is a high performing approach, as it achieved an average AUC of 0.9383. This result is the best result, as the dataset used for training and testing the model has the highest rate of 41 features. Secondly, the data used in this work equally considered a different kind of cyber-attacks, where buffer overflow is 0.8486, ipsweep is

0.9999, normal packet is 0.9824, rootkit is 0.8012, sql attack is 0.9977, and worm reported 1.00 in AUC performances. 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

Alshehri A., Khan N., Alowayr A., & Alghamdi M., (2022) Cyberattack Detection Framework Using Machine Learning and User BehaviorAnalytics.Computer Systems Science & Engineering DOI:10.32604/csse.2023.026526.

Arabo, A., &Pranggono, B. (2013). Mobile malware and smart device security: Trends, challenges and solutions. In Control Systems and Computer Science (CSCS), 2013 19th International Conference on (pp. 526–531). IEEE.

Azeez N., Taiwo O., Chioma I., & Abidoye A., (2023) Cyber Attack Detection in a Global Network Using Machine Learning Approach. FUOYE Journal of Engineering and Technology, Volume 8, Issue 4, December 2023http://doi.org/10.46792/fuoyejet.v8i4.1113

Bebeshko B., Khorolska K., Kotenko N., Kharchenko O., &Zhyrova T., (2021) Use of Neural Networks for Predicting Cyberattacks. CEUR Workshop Proceedings (CEUR-WS.org)Cybersecurity Providing in Information and Telecommunication Systems

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. 10.59200/ICARTI.2023.026

Bocu R., &Iavich M. (2023). Real-Time Intrusion Detection and Prevention System for 5G and beyond Software-Defined Networks. Symmetry, 15, 110. https://doi.org/10.3390/sym15010110

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/

Cao J., Ma M., Li H., Ma R., Sun Y., Yu P., & Xiong L. (2019). A survey on security aspects for 3GPP 5G networks. IEEE Communications Surveys & Tutorials, 22(1), 170-195. https://doi.org/10.1109/COMST.2019.2933831

Dhanya K., Sulakshan V., Kartik S., Anjali T., Senthil T., & Kumar T., (2023) Detection of Network Attacks using Machine Learning and Deep Learning Models. International Conference on Machine Learning and Data Engineering. Data Engineering 10.1016/j.procs.2022.12.401

Kravchik M., & Shabtai A., (2018) Detecting Cyberattacks in Industrial Control Systems Using Convolutional Neural Networks. arXiv:1806.08110v2 [cs.CR] 10 Dec 2018

Lai, C., Lu, R., Zheng, D., & Shen, X. (2020). Security and privacy challenges in 5G-enabled vehicular networks. IEEE Network, 34(2), 37-45.

Liu, H., & Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. Applied Sciences, 9(20), 4396.

Maksim Iavich, Giorgi Iashvili, ZhadyraAvkurova, SerhiiDorozhynskyi, and AndriyFesenko (2021)"Machine Learning Algorithms for 5G Networks Security and the Corresponding Testing Environment" CPITS-II-2021: Cybersecurity Providing in Information and Telecommunication Systems, pp.139-149

Mantas, G., Komninos, N., Rodriguez, J., Logota, E., & Marques, H. (2015). Security for 5G Communications. John Wiley & Sons, Ltd.

Mijumbi, R., Serrat, J., Gorricho, J., Latré, S., Charalambides, M., & Lopez, D. (2016).

Management and orchestration challenges in network functions virtualization. IEEE Communications Magazine, 54(1), 98-105.

Neha Yadav, SagarPande, 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

Ogbeta L.K., and Nwobodo Lois (2023). Neuro based strategy for real time protection of wireless netork ecosystem against DDOS attack. [J] I1SRED; ISSN 2581-7175; pp. 79-98.

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); Volume 2, Issue II, February 2023, No. 38, pp. 390-402

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

Prakash R., 7 Rajeshwari K., (2024) Using Machine Learning to Detect Cyber Attacks. International Journal of Research Publication and Reviews, Vol 5, no 2, pp 2793-2806

Salahdine, F. (2018). Compressive spectrum sensing for cognitive radio networks. arXiv preprint arXiv:1802.03674

Salahdine, F., &Kaabouch, N. (2019). Social engineering attacks: A survey. Future Internet, 11(4), 89.

ShariatM., Bulakci O., Domenico A., Mannweiler C., Gramaglia M., Wei Q., Gopalasingham A., Pateromichelakis E., Moggio F., Tsolkas D., Gajic B., Crippa M., &Khatibi S., (2019). A Flexible Network Architecture for 5G Systems.Hindawi Wireless Communications and Mobile Computing Volume 2019, Article ID 5264012, 19 pages https://doi.org/10.1155/2019/5264012

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., &Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from over-fitting. Journal of Machine Learning Research, 15(1), 1929-1958.https://www.cs.toronto.edu/~hinton/absps/JMLRdropout.pdf

Tata, C., &Kadoch, M. (2022). Network Coding-Based D2D Transmission for Public Safety Networks over LTE HetNets and 5G Networks. Hindawi Wireless Communications and Mobile Computing, 2022, Article ID 8889718. https://doi.org/10.1155/2022/8889718

Tian Z., Sun Y., Su S., Li M., Du X., and Guizani M., (2019). Automated attack and defense framework for 5g security on physical and logical layers. arXiv preprint arXiv:1902.04009, 2019.

Tomida S., Mizutani K., & Harada H., (2021). Radio Protection Area Estimation Methods for Spectrum Sharing-based 5G System. 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)2021 IEEE | DOI: 10.1109/CCNC49032.2021.9369627

Tran T., Hajisami A., Pandey P., Pompili D., (2017). Collaborative mobile edge computing in 5G networks: new paradigms, scenarios, and challenges. IEEE Commun Mag. 2017;55(4):54-61.

Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2017). Understanding deep learning requires rethinking generalization. arXiv preprint arXiv:1611.03530. https://arxiv.org/pdf/1611.03530.pdf