*International Journal Of Engineering Research And Development*
*e- ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com*
*Volume 21, Issue 4 (April 2025), PP 59-66*

# A Deep learning based Network Intrusion Detection System

[1]FAGBOHUNMI, Griffin Siji [2]Uchegbu Chinenye E.
[1]*Department of Computer Engineering Abia State University, Uturu, Abia State, Nigeria*
[2]*Department of Electrical and Electronics Engineering Abia State University, Uturu, Abia State, Nigeria*

***Abstract***
*In recent times, there has been an increase in the connectivity of different types of endpoint devices, this is possibly due to the widespread use of IoT applications and embedded systems. This increase in connectivity has led to ease of communication with many devices, such as home appliances, office and industrial equipment. The dark side of this enhanced connectivity is the widespread of adversarial threats on endpoint devices. Hackers exploit vulnerabilities in the security architectures of the communication protocols used by these devices to compromise their communication efficiency. There is therefore the need to devise intrusion detection systems to combat this menace. Intrusion detection systems in this context use cyber-security mechanisms to prevent compromise to communication between endpoint devices. In this paper, an intrusion detection system using deep machine learning is proposed. The system employs an adaptive and difficult to break architecture to prevent intrusion into endpoint devices network by classifying network attacks into distinct categories. The aim here is to develop a deep learning network that provides an adaptive Intrusion Detection System (IDS) capable of learning the properties of known network threats and be able to build on this knowledge to devise means of identifying yet to be recognized network threats using their behavioural patterns. The dangers of adversarial threats will then be greatly minimized. To validate the network model, the CIC-IDS-2018 dataset was used because its knowledgebase contains features to identify current communication patterns using its automated attack scenarios. Results from simulation experiments showed improvement in detecting accuracy compared to some state of the art intrusion detection techniques.*

***Keywords****: Cybersecurity; Zero-day attacks; Deep learning; Intrusion detection systems*

---

Date of Submission: 02-04-2025 Date of acceptance: 12-04-2025

---

## I. Introduction

In recent times, there has been an increase in the connectivity of different types of endpoint devices, this is possibly due to the widespread use of IoT applications and embedded systems. The connectivity has led to ease of communication with many devices, such as home appliances, office and industrial equipment. These connectivity among endpoint devices has led to different ways of doing things and seemingly over reliance on ICT which has undoubtedly revolutionized how we relate with various devices in our daily lives. Another positive side to this ICT enhanced environment is the possibility that organizations can have real time communication using several devices to better .scale up network access. These communications has also led to real time connectivity among various organizations fostering enhanced collaboration amongst them for better service delivery to customers [1].

It should be noted that the possibility of digital information transfer amongst several networks has resulted in the possibility of network intrusion by adversarial agents. This is achieved by exploiting vulnerabilities in the protection mechanisms in the communicating devices which is not always strong to deter network intrusion. It is therefore expedient that an effective intrusion detection system be deployed to protect endpoint devices on the internet from being compromised through attack on their confidentiality, integrity and availability [2]. Defensive mechanisms against attack vectors on networked devices are always deployed in a layered architecture and the first layer deals with the control of network security which is generally referred to as the Intrusion Detection System (IDS). An IDS periodically checks the network traffic to detect and report a compromise using comparison with the different levels of a customized previously configured system. It is necessary that the detection should be made before any damage is done to the data on the network.

The algorithm used by IDS is based on the fact that the characteristic feature of a network traffic that has been compromised is usually different from that of a valid network traffic feature, an IDS therefore uses these difference in characteristic features to isolate network traffic with adversarial tendencies. However in many instances a clear cut difference cannot be made between normal network feature and an adversarial threat feature, in this case, an intelligent intrusion system using machine earning is employed [3].

The different types of intrusion detection system includes, (i) Network Intrusion Detection system (NIDS), which comprises of the hardware (sensors) and the software (console). They are both used to monitor, detect and control the features of network traffic packets among the communicating devices for possible intrusion or abnormal network traffic patterns. (ii) Host Intrusion Detection System (HIDS). HIDS is normally resident on a computer server and it monitors, detects and control network traffic on that system network alone. The HIDs being specific to a particular network system is usually more effective than the NIDS because it is capable of accessing encrypted messages going on in the network, which includes system configuration, registers, network files and database attributes. (Iii) A Cloud Intrusion Detection System (CIDS) comprises of the cloud network and host layers. The function of the cloud layer is the provision of secure authentication for all access into the system's network comprising of a shared group or application programming interface. It also differentiates between existing IDS and hypervisors. In order to study the flow of traffic in the network, IDS employs different detection mechanisms, these include:

- **Signature based detection**: This is also known as knowledge based detection as it uses stored database to identify network patterns that matches existing or already known signatures. It uses the knowledge from a known attack corresponding to a particular signature to detect anomaly in a network pattern. It however requires repetitive updates to identify yet to be identified attack patterns.
- **Anomaly based detection**: This is also referred to as behaviour based detection, here the network traffic is examined to detect patterns that are not in consonant with the normal or stored traffic characteristics in the system's database. Anomaly based detection uses statistical techniques to sample network traffic in order to identify difference between the obtained dataset value and the expected one. Once the difference exceeds a threshold value, the administrator will be alerted of a deviation from the norm. Anomaly based detection is capable of detecting new anomalies unlike the signature type, however this is achieved at a higher processing power especially if the deviation occurs for a long duration. It also suffers from false positives where a little deviation from a normal traffic dataset results in an alert to the administrator.
- **Stateful protocol analysis**: This type of intrusion detection performs a comparison with known protocol in order to detect anomaly. It employs the already known dataset vector profile to find out sequences of random commands in the application and network layer of the communication stack.

It should be noted however that every detection method cannot distinctively differentiate between absolute good network patterns and a compromised network patterns. This invariably leads to the problem of false positives and false negatives. The time used by the intrusion detection system to send alerts for wrong dataset classification leads to decreased CPU's efficiency. In order to come around these shortcomings, machine learning based intrusion detection system has been proffered by some researchers. Machine learning techniques such as Decision trees, Naïve Bayes, Support Vector Machines can be used to develop an intelligent intrusion detection system with high degree of accuracy. The shortcoming of these machine learning techniques is the large datasets needed to be trained by the machine learning model employed. Many machine learning techniques that uses unsupervised learning methods fail to get optimal performance in multiclass network traffic with large datasets [4].

Currently there is the self-learning intrusion detection systems capable of detecting yet to be identified abnormal network traffic which can cause intrusion to the system. This type of intrusion detection has the capability of learning the properties of network datasets which can be classified as either normal or abnormal traffic, in this way the identity of any intrusion can be determined using this machine learning techniques. An ideal machine learning technique that can solve the problem of correctly identifying the status of any intrusion is deep learning. This is because of its ability to be effective when very large datasets are involve, it has been successfully used in image processing, speech recognition and natural language processing, just to mention but a few [5].

In this paper, deep learning algorithms will be used to design an intrusion detection system that can correctly predict the status of any network datasets irrespective of its size. The deep learning model used in this paper is based on the CIC-IDS-2018 dataset intrusion behavioural pattern which is based on a realistic network attack model [6]. The rest of the paper is as follows, section 2 deals with related works, section 3 discusses the model approach used in this paper, section 4 describes the model in detail while section 5 concludes the paper with a mention on future directions.

## II. Related Works

The idea behind network intrusion system is that hackers tries to exploit the weakness in the security architecture of network devices most of which do not have adequate network security. probably due to their embedded nature. Through these weak links intruders (hackers) can penetrate a network system. Hackers can try to access a network remotely through weak endpoint devices in order to compromise the network operations. Hackers tend to gain through the process of intruding into a system to get sensitive information required for the proper functioning of the network. This on the other hand causes huge loss to the company using the network. An intrusion detection system (IDS) is designed to detect these intrusions and prevent their operations by means of

analysing network traffic patterns. There are different techniques used by IDS, this includes filtering the network traffic in order to classify normal and abnormal network traffic datasets, however the process required in classifying network dataset can lead to a reduction in network's throughput. This then calls for technique which can correctly detect network intrusions whether having signature or not as well as not reducing the network's efficiency. A machine learning technique referred to as deep learning is an ideal candidate for the capability mentioned stated here.

Deep neural networks is made up of continuous multi-layer extraction technique which is ideal for systems having large size of data [7]. In current work on IDS, there has been an increase in accuracy to the detection of network traffic anomaly however the lack of an open source datasets makes objective comparison impossible. This led to researchers using simulation of network traffic with real life feature to validate intrusion detection systems algorithms. The first work on network traffic IDS datasets used the KDDCup, this research was a DARPA project. The KDDCup dataset was developed in Lincoln lab and it used the tcpdump that is based on a closed network generated traffic with the inclusion of manually injected attacks for the simulation of US Air Force bases. Now a situation where the closeness to real life for this simulation model is required has led to the design of the NSL-KDD dataset [8]. The NSL-KDD dataset though provided reduction in redundancy does not still provide a good representation of a real life network traffic dataset, however it is the most referenced dataset used by researchers on IDS. Another dataset used currently is the CIC-IDS-2018 which was designed by Moustafa and Slay [6, 9] during their research at the University of New South Wales. The dataset used by these researches was duplicated to the KDDCup dataset by means of Association Rule Mining for selecting the features, however only few standard features was included in the dataset, this made comparison and validation with other datasets difficult. Nine classes of attacks were identified by the CIC-IDS-2018 [5] these attacks represented common vulnerabilities and exposures (CVE). CVE represents a library of simulated attacks used by most IDS datasets [10]. CIC-IDS-2018 datasets has more than forty features used in the representation of real network traffic, out of which only nine new attack classes were stressed upon [11] Designers has come up with a reduced training sets from the over 2.6 million records available used in some IDS models. It should be noted that only a subset of the available 2.6 million records are used in most IDS models, this causes a reduction in accuracy of the classification which eventually leads to false positives and false negative alarms. Now with the use of deep learning neural network, all 2.6 million records in the CIC-IDS-2018 can be trained with the use of classification techniques such as data cleansing, editing and reduction to produce a reduced classification that is representative of the entire dataset records.

The datasets first used by researchers for IDS machine learning was the original DARPA datasets using machine learning techniques such as Decision Trees, Support Vector Machines, Random forest etc. These machine learning techniques are not as scalable as the deep neural network and as IDS datasets usually contains large database, most of the earlier machine learning techniques will not be appropriate as some of them may not be accurate and not be able to give a good representation of the extensive dataset records and attack types contained in CIC-IDS-2018 [12, 13].

For the CIC-IDS-2018 dataset used in this paper, it uses binary classifiers to determine the status of an instance of traffic pattern, as being normal (benign) or abnormal (malignant). In literature, researchers usually design two models to determine the status of an instance either as binary or categorical classification. The CIC-IDS-2018 dataset also contains the KDDCup dataset as a sub-dataset with 24 attack types. According to the researchers in [14], the imbalance in the dataset record being represented in a model can be improved by categorizing datasets for training and testing separately. In this way more datasets can be involved in the model design. Usually these two categories of datasets are merged while sampling is done to decide which datasets are to be used for training and testing respectively, hence the LaNet-5 model was used by these researchers to separate the network threats or the NSL-KDD dataset. From the results given by these researchers, the LaNet-5 model has improved performance both in range of attack types isolated as well as in threat detection accuracy

Researchers in [15] used time series modelling which was made up of a combination of both CNN and recurrent neural network techniques A tensorflow engine using Nvidia graphics processing unit was the hardware used for the simulation, it also included the KDDCup dataset. Researchers in [16] designed an intrusion detection system using hyper-parameter tuning involving a host of network topologies. Though the sampling used in their design did not give a good representation of all network traffic dataset their results outperformed the compared models used in their work. The researchers in [16] designed a deep neural network {DNN) based intrusion detection system, where it was adduced that the ReLU activation function used did not suffer the vanishing gradient effect noticed in other designs. Their work was an improvement over other nonlinear activation functions. It should be noted that, though a lot of network traffic datasets are used in literature, CIC-IDS-2018 gives a more comprehensive representation of the full range of network traffic datasets, therefore gives a better approximation to a real network scenario. It was shown by researchers in [17], that CIC-IDS-2018 dataset gave a 79% accuracy for binary classification and 66% accuracy when multiclass classification model was used. Binary classification is best suited for a single hidden layer deep neural network, while the multiclass classification works well with deep neural network with two to three hidden layers. In the work of researchers in [18], deep neural network with

more than three hidden layers resulted in reduction in performance when the hyper-parameter model was used. Researchers in [19] designed a deep learning model using 10 hidden layers having 10 neurons per layer Validation was performed using three separate experiments with ten iterations each The aim of the first experiment was to determine an optimum activation function to be used for the ten-fold iterations used in their work. The second experiment on the other hand was to determine the features present in the datasets, while the third experiment was to combine the parameters in the first two experiments to compare accuracy in intrusion detection among the activation functions. Their work did not however specify the type of classification model nor the architecture used. In the work of researchers in [20], a multiclass classification model with five hidden layers in a deep neural network for CIC-IDS-2018 was investigated. Their design showed an improvement of 87%, 91% and 95% respectively when compared with work of researchers in [21-23]. The deep random neural network proposed in [22], also gave a 96% intrusion detection accuracy while an accuracy of 97% was obtained for binary representation

## III. Materials and Methods

In this paper, a deep learning model using CNN, and a regularized multilayer perceptron was designed. The regularization was done using different values for the regularization parameter alpha (α). Regularization is needed here so as to prevent overfitting. Hence the value for regularization to the deep neural network model helps to minimize variance in the model output so that accurate intrusion status can be predicted for the network traffic datasets. CNN is opposed to the fully connected feed forward neural network (FNN) used in some literatures. In contrast to FNN, CNN deploys convolution instead of multiplication or dot operator used in FNN for its mathematical operation. Convolution operator as used in this paper uses some hyper-parameters such as filter dimension, filter amount and strides for the generation of its matrix output. Also in order to mitigate the diminishing tensor dimensions resulting from the propagation of the input through the many convolutional layers, padding of output was used. An extra layer was used in between convolutional layers in order to minimize the effect of network traffic not represented in the CIC-IDS-2018 dataset. Lastly a fully connected multilayer perceptron model was designed and then the classification of the output layer was done. As said earlier, the CIC-IDS-2018 dataset was used in this paper due to its closeness to the compromise that can be inflicted on real world network traffic and the extensive number of datasets represented by it. The CIC-IDS-2018 dataset was used with some models proposed in literature and the result was not optimal, meaning a more optimal yielding model must be sought.

It should be noted here that there are over two million raw dataset instances, however research has shown that only eleven attack families are used for dataset training and testing. The attack families and its abridged explanation is shown in table 1.

Table I. Attack categories and descriptions

| Attack | Short Description |
|---|---|
| Normal | Good network traffic |
| Fuzzers | Corrupt traffic due to insertion, scanning of port or virus |
| Analysis | Corrupt traffic due to data interception or relay attack |
| Backdoor | Attack that overrides network security procedures |
| DoS | Denial of service due to preventing data to reach authorized node. |
| Exploits | Gaining access to network through OS firewall weakness |
| Generic | Attack due to brute force and continuous cryptographic key insertion |
| Reconnaisesance | Vulnerabilities or weakness is checked for in target system |
| Shellcode | Malignant codes are inserted onto the network |
| Worm | Malignant codes are replicated onto the network using node mirroring |

The model used in this paper makes use of the keras library acting as a prototype for the tensorflow framework. Keras is ideal in this work because of its easy to use high level application programming interface which makes it easy to be used by people with limited experience in simulation. Also tensorflow supports large scale and enterprise level application and it's therefore suited for deep leaming models which involves high volume of data. Additionally the framework contains adequate tools that can be easily modified with compatibility with many deep learning architecture such as CNN and RNN. It can also be easily adapted on Central Processing Unit and Graphics Processing unit [18]. Training of datasets was done on Google Colab, which is a free cloud based environment for the training of machine learning or deep learning models. The deep learning model in this paper was trained on GPU framework that is enabled on Google Colab so as to enhance dataset optimization.

The intrusion detection system network dataset calls a pre-processing function in order to convert its class into numeric vectors. A technique of data encoding was used to convert dataset class to numeric vectors, this was also necessary to detect unique feature category for unidentified data. Normalization was performed on the feature categories for use on CNN input, the labels were encoded using one-hot representation to denote the number of classes. Ten classes were identified for the multiclass model, where eleven was used to represent eleven

categories of attacks, the remaining one represented a normal flow of traffic. The model includes an optimization of semi-dynamic hyper-parameter to determine the following (i) number of iterations in the experiment (ii) determination of learning rate (iii) optimization algorithm and (iv) batch size.

The optimization was started midway from grid search as used in the hyper-parameter optimization used by researchers in [24] to determine a baseline. After the baseline was determined, the corresponding hyper-parameters are then generated by moving into the trajectories of the search space. Whenever any hyper-parameter has better representation of the network traffic dataset than the baseline, it replaces the baseline model, this substitution continues until a decline is experienced in the performance of successive hyper-parameter selection.

Apart from the tuning done to the hyper-parameter to obtain an optimum deep neural network model, some functions from the tensorflow framework such as Call_Back, Early_Stopping and Model_Checkpoint were invoked to decrease the time required to iterate through the search space before the model converges to the optimum hyper-parameter value. The pseudocode in figure 1 shows the semi-dynamic approach used in traversing along the sample space for the dropout rate, batch size and the learning rate.

Function Optimization $F_j(f_1 \ldots\ldots f_j)$
Optimization Algorithm for Gradient descent $G_m(g_1 .g_m)$
Rate of dropout $D_t(d_1 ..d_t)$
Size of Batch $B_x$ $(b_1 \ldots.b_x)$
Rate of learning $L_k$ $(l_1 \ldots l_k)$
Arrange the dropout rate in ascending order
for m = 1 to j
if Dm < Dm+1
write Dm
else write $D_{m+1}$
next m
end m
Assign the mid value dropout batch size value and learning rate value within the dropout range
Int(j/2) // D varies from 1 to j now sorted
Dmid = $D_{j/2}$ // mid value for dropout
Bmid = $B_{j/2}$ // mid value for batch size
Lmid = $L_{j/2}$ // mid value for learning rate

Fig. 1. A semi-dynamic hyper=parameter optimization approach

It should be noted that it is possible to get the model of the architecture proposed in this paper through trial when combined with optimization of the hyper-parameter. The architecture is made up of a double stacked convolutional layer, Max Pooling and Dropout. The double stacked convolutional architecture was included to enhance accuracy of the intrusion detection, while Dropout after each Max Pooling layer aids in the reduction of overfitting.

In figure 2, the classification architecture proposed in this paper is shown with dropout in the hidden layers while an increase in dropout was done with more hidden layers. This is done so as to regulate the parameter values for overfitting. The hidden layers makes use of a combination of logistic and then the softsign was used as its activation function before generating the output layer.

| |
|---|
| Convolut input layer: convolut |
| Maximum poolID 1: MaxppolID |
| ConvolutID: convolute |
| ConvolutID 1: convolute |
| Maximum poolID 1: MaxppolID |
| Convolut ID 2 : convolute |
| Convolut ID 3 : convolute |
| Convolut ID 4 : convolute |
| Convolut ID 5 : convolute |
| Maximum poolID 2: MaxppolID |
| Dropout 2: drop |
| Flat: convolut1D |
| Dense: Dense |
| Drop0ut 3: MaximumpoolD |
| Dense_1 :Dense |

Fig. 2. Proposed classification architecture

In most cases considered in literature, class differences does not affect the performance for a binary model, this has also proven to be the case here where the ten class model depicted in this paper results in a balance classification having 29k instances in the upper class and 28k in the lower class. This can be attributed to the activation functions used in the paper which was the logistic and softmax activation functions. This is opposed to

the ReLu and softmax activation functions used by researchers in [25]. The model in [25] cause imbalance in the lower and upper divisions of classes. This imbalance causes an increase in the convergence time for the model, as bootstrapping was required to add to the under-sampled classes, also a second dataset was added which involves merging of the unbalanced classes and modulating it to present a somewhat equilibrium status for all the classes. This causes the model to be complex in comparison to the model proposed in this paper.

## IV. Results and Discussions

The semi-dynamic hyper-parameter tuning was combined with the model used in this paper to train the network traffic datasets using both the already partitioned CIC-IDS-2018 dataset and the user defined CIC-IDS-2018 dataset.

### A. Testing the Model with the Already Partitioned CIC-IDS-2018 Dataset

According to the researchers in [17], [22], the benchmark for an already partitioned CIC-IDS-2018 dataset having the full complement of features on deep learning models is from 78.47% to 98.5%. In the model proposed in this paper, which consists of semi-dynamic hyper-parameter tuning, results in considerable improvement over the other state of the art models used in comparison. On the average, the deep learning model has a 95% accuracy with the CIC-IDS-2018 dataset. In order to prevent overfitting with the intrusion detection, the categorical cross entropy was combined with the Adam optimization algorithm. The learning rate of the model was 0.002 resulting in a reduced dropout rate for the upper and lower layers of the dataset. The graph for the accuracy of the learning curve for the multiclass classification model is shown in figure 3. Figure 4 shows the rate of detection for the various categories of attacks.

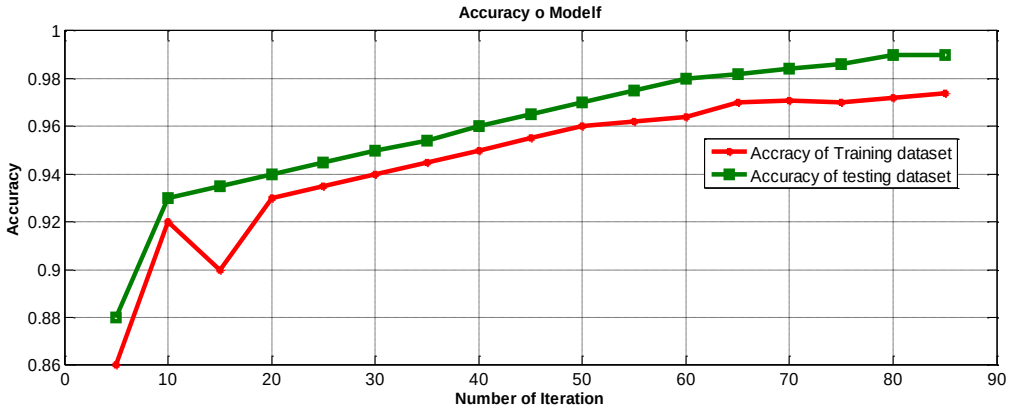


Fig. 3. Accuracy of Learning curve using multiclass classification mode

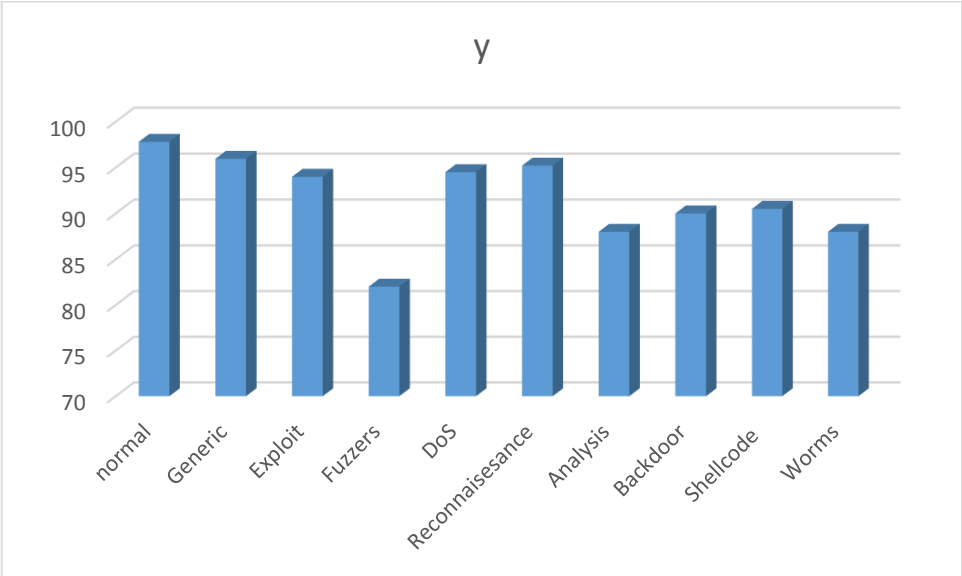


Figure 4 Rate of Detection for the different types of attacks

From figure 4 it will be noticed that the rate of detection for the classes not incudes in the CIC-IDS-2018 dataset was very much lower than those included in the dataset.

**B. Testing the Model with the user-defined CIC-IDS-2018 Dataset**

In addition to the CIC-IDS-2018 dataset used in literature, a user-defined CIC-IDS-2018 dataset was also used in this paper. This is necessary considering the attack variants in the case study which is the commercial banks in Nigeria. The user-defined UNSW-N15 dataset was able to mirror the attack variants by hackers on Nigerian bank's database.

For the user-defined partitioned CIC-IDS-2018 dataset, the benchmark of its full complement was considerably higher than that of already partitioned CIC-IDS-2018 dataset. The model presented in this paper with the hyper-parameter counterpart produced 96.2% for the 35% testing dataset. As explained earlier, overfitting of network intrusion dataset accuracy was prevented using the categorical cross entropy with Adam optimization algorithm. With this approach, the learning rate of the deep neural network model was 0.007 with same dropout rate as in the already partitioned CIC-IDS-2018 dataset. The accuracy of the learning curve with the multiclass classification model is shown in figure 5. Figure 6 shows the rate of detection for the categories of attacks considered. Like in the case of the already partitioned CIC-IDS-2018 dataset, the rate of detection for the classes not included in the user defined UNSW-MB15 dataset was considerably lower than those represented in the dataset.

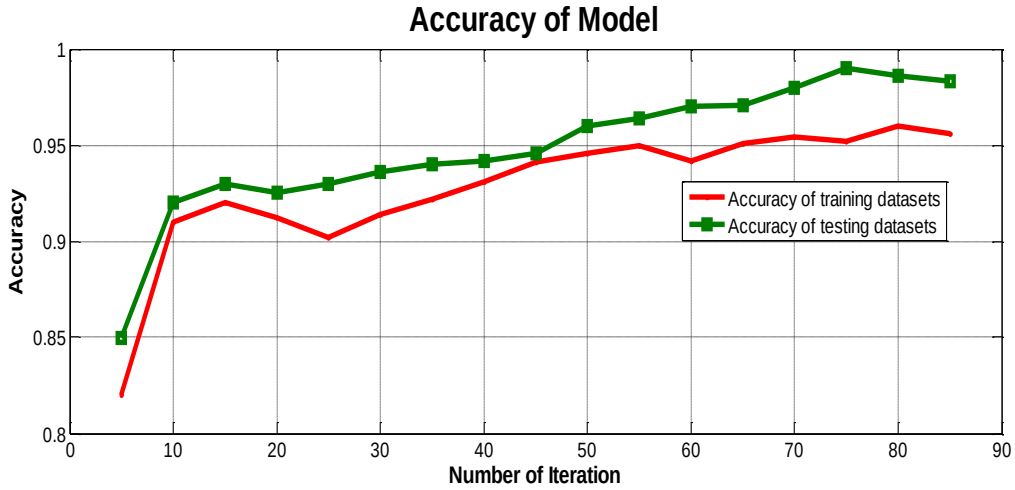


Figure 5 Accuracy of Learning curve using the multiclass classification model

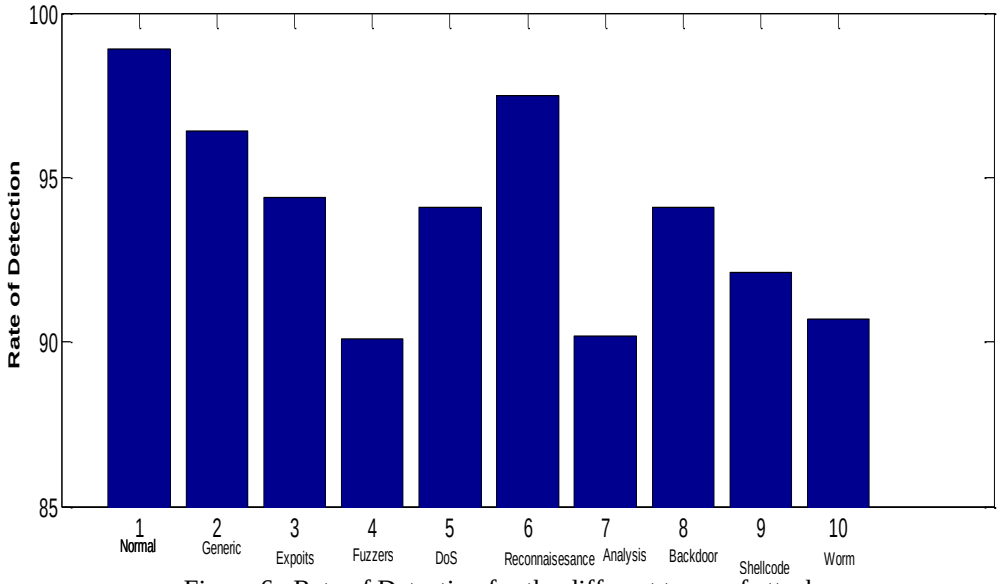


Figure 6 Rate of Detection for the different types of attacks

## V. Conclusion

In this paper, network intrusion detection system was discussed with current CIC-IDS-2018 dataset which is the most widely used network traffic model used currently in literature. The CIC-IDS-2018 dataset has most features of network threat in addition to common cybersecurity threats. The deep learning classification architecture together with the semi-dynamic hyper-parameter tuning used in this paper shows considerable better results for multiclass models in comparison to that of other deep learning based intrusion detection system models.

The model presented in this paper, has an overall accuracy of 96.2% and 97% for already partitioned CIC-IDS-2018 dataset and the user defined CIC-IDS-2018 dataset respectively. Even though the model presented in this paper has shown considerable better results when compared to other state of the art deep neural network models, there is still room for improvement based on the idea presented in this paper where feature reduction techniques was used to develop the user defined CIC-IDS-2018 dataset It is therefore recommended that variants of the CIC-IDS-2018 dataset be developed for different network environment, such as academic institutions, chemical industries, automobile industries just to mention but a few. Future research works should also be capable of detecting zero day attacks, i.e. attacks variants that are occurring in real time.

## References

[1] Ashiku, Lirim, and Cihan Dagli. (2019) "Cybersecurity as a Centralized Directed System of Systems using SoS Explorer as a Tool." In 14th Annual Conference System of Systems Engineering (SoSE), pp 140-145. IEEE.

[2] Duque, Solane, and Mohd Nizam bin Omar. (2017) "Using data mining algorithms for developing a model for intrusion detection system (IDS)." Proceedings in Computer Science Journal, Vol 61 pp 46-51

[3] Stallings, William, Lawrie Brown, Michael D. Bauer, and Arup Kumar Bhattacharjee.(2015) Computer security: principles and practice. New Jersey, USA. Pearson Education Publisher.

[4] Shone, Nathan, Tran Nguyen Ngoc, Vu Dinh Phai, and Qi Shi. (2018) "A deep learning approach to network intrusion detection." In IEEE Transactions on Emerging Topics in Computational Intelligence, Vol 2, no. 1 pp 41-50.

[5] Lipton L, Zachary C., John Berkowitz, and Charles Elkan. (2018) "A critical review of recurrent neural networks for sequence learning." arXiv preprint arXiv:1506.00019.

[6] Moustafa, Nour, and Jill Slay. (2019) "CIC-IDS 2018: a comprehensive data set for network intrusion detection systems (CIC-IDS 2018 network data set)." In IEEE 2019 military communications and information systems conference (MilCIS), pp 1-6.

[7] Osken, Sinem, Ecem Nur Yildirim, Gozde Karatas, and Levent Cuhaci. (2022) "Intrusion Detection Systems with Deep Learning: A Systematic Mapping Study." 2022 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT), pp 1-4.

[8] McHugh John. (2006) "Testing intrusion detection systems: a critique of the 2004 and 2005 darpa intrusion detection system evaluations as performed by lincoln laboratory." ACM Transactions on Information and System Security (TISSEC) Vol 3, no. 4: pp 262-294.

[9] Moustafa, Nour, and Jill Slay. (2018) "The significant features of the UNSW-NB15 and the KDD99 data sets for network intrusion detection systems." In 2018 6th international workshop on building analysis datasets and gathering experience returns for security (BADGERS), pp 25-31.

[10] "Home." CVE. Accessed December 12, 2019. http://cve.mitre.org/about/index.html.

[11] Janarthanan, Tharmini, and Shahrzad Zargari. (2020) "Feature selection in UNSW-NB15 and KDDCUP'99 datasets." In 2020 IEEE 26th International Symposium on Industrial Electronics (ISIE), pp 1881-1886.

[12] Pfahringer Bernhard.(2007) "Winning the KDD99 classification cup: bagged boosting." SIGKDD explorations, vol 1, no. 2: pp 65-66.

[13] Sabhnani Maheshkumar, and Gürsel Serpen. (2009) "Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset within Misuse Detection Context." MLMTA, pp 209-215.

[14] Lin, Wen-Hui, Hsiao-Chung Lin, Ping Wang, Bao-Hua and Jeng-Ying Tsai. (2022) "Using convolutional neural networks to network intrusion detection for cyber threats." 2022 IEEE International Conference on Applied System Invention (ICASI), pp 1107-110.

[15] Vinayakumar, R., Soman K. P, and Prabaharan Poornachandran. (2020) "Applying convolutional neural network for network intrusion detection." In 2020 IEEE International Conference on Advances in Computing, Communications, and Informatics (ICACCI), pp 1222-1228.

[16] Vinayakumar R., Mamoun Alazab, Soman K. P, Prabaharan Poornachandran, Ameer Al-Nemrat, and Sitalakshmi Venkatraman. (2022) "Deep Learning Approach for Intelligent Intrusion Detection System." IEEE Access Vol 7: pp 41525-41550.

[17] "Keras: The Python Deep Learning Library." (2023) n.d. Home - Keras Documentation. Accessed January 11, 2024. https://keras.io/.

[18] Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. (2019) Deep learning. MIT press.

[19] Zhiqiang Liu, Ghulam Mohi-Ud-Din, Li Bing, Luo Jianchao, Zhu Ye, and Lin Zhijun. (2022) "Modeling Network Intrusion Detection System Using Feed-Forward Neural Network Using CIC-IDS 2018 Dataset." 2022 IEEE 9th International Conference on Smart Energy Grid Engineering (SEGE), pp 299-303.

[20] Latif Shahid, Zeba Idrees, Zhuo Zou, and Jawad Ahmad. (2023) "DRaNN: A Deep Random Neural Network Model for Intrusion Detection in Industrial IoT." 2023 IEEE International Conference on UK-China Emerging Technologies (UCET), pp 1-4.

[21] Kasongo Sydney Mambwe and Yanxia Sun. (2023) "A deep learning method with wrapper based feature extraction for wireless intrusion detection system." In Journal of Computers & Security Vol 92: pp 101- 112.

[22] Choudhary, Sarika, and Nishtha Kesswani. (2022) "Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 Datasets using Deep Learning in IoT." Proceedings in Computer Science conference, Vol 167 pp 1561-1573.

[23] Tran Viet, Khoa, Saputra Yuris Mulya, Hoang Dinh Thai, Trung Nguyen Linh, Diep Nguyen N, Ha Nguyen Viet, and Dutkiewicz Eryk. (2022) "Collaborative learning model for cyberattack detection systems in IoT industry" 2022 IEEE International Conference.

[24] Paul Supratik, Vitaly Kurin, and Shimon Whiteson. (2022) "Fast efficient hyper-parameter tuning for policy gradients." arXiv preprint arXiv:1902.06583.

[25] Lirim Ashiku and Cihan Dagli (2021) "Network Intrusion Detection System using Deep Learning" Complex Adaptive Systems Conference in Computer Science Vol 185 pp 239–247