IJCRSSL
International Journal of Current Researches in Sciences, Social Sciences and Languages

Volume: 04 | Issue: 02 | 2024 | Open Access | Impact Factor: 5.735

An Improved Intrusion Detection System Using Densenet

Yusuf Musa Malgwi[1], Simon Shonva Wagbe[2], Sani Dan Azumi Abdulkadir[3], Lydia Chilaodiri Okpalaifeako[4]

[1]Department of Computer Science, Modibbo Adama University Yola, Adamawa State
[2]Department of Mathematical Sciences, Faculty of Science, Taraba State University
[3]Department of Mathematical Sciences, Faculty of Science, Taraba State University
[4]Department of Computer Science, Federal Polytechnic Bali, Taraba State

Abstract: As the quantity of devices connected to the internet rises, safeguarding against intrusions becomes increasingly imperative. However, due to the regular emergence and evolution of malicious threats, networks require a sophisticated security solution. This study proposes the utilization of a deep learning model, employing DenseNet, to construct an enhanced intrusion detection system that exhibits both effectiveness and intelligence. In order to enhance the effectiveness and predictability of the intrusion detection system, the convolutional neural network conducts convolutions to gather local features and extract them within the proposed model. Experiments utilizing publicly available datasets, namely CSE-CIC IDS2017 and CSE-CIC IDS2018, are carried out to evaluate the efficacy of the proposed system. The research findings indicate that the proposed system surpasses existing intrusion detection approaches, achieving an average accuracy of 96.9% and 99.7% respectively in identifying malicious attacks. Ultimately, the accuracy and detection rate affirm the detection effectiveness of the DenseNet model, which has been observed to yield favorable outcomes in intrusion detection. Further investigation is warranted for this model to explore additional parameters, thereby enhancing its performance and fortifying its defenses against attacks.

Keywords: Intrusion, Deep Learning, Convolutional Network, DenseNet, Anomaly Detection

INTRODUCTION

Recently, the use of Internet is growing rapidly. By January 2020, approximately 4.54 billion individuals had started utilizing the internet (Clement, 2020). An enduring challenge in the realm of intrusion detection, noted since the inception of IDS technology, is the False Positive Rate as highlighted (Lang & Lui 2020). This issue has been a focal point for researchers, as it places considerable strain on security analysts owing to the prevalence of false alarms. This strain can inadvertently result in the oversight of critical cyber threats. To address this challenge, ongoing enhancements to IDS systems are imperative, particularly as network landscapes evolve continuously. Consequently, as networks undergo transformations, novel forms of intrusions inevitably emerge, necessitating continuous refinement of IDS capabilities.

For each kind of invasive behavior, two types of intrusion detection exist: both intrusion detection systems derived from the host and the network (Mishra et al., 2018). A system for intrusion detection employing the use of network activity to identify threats is referred to as a "Network Intrusion Detecting System." Mirroring network equipment like network switches, routers, and monitoring devices collects and analyzes network behaviors in order to detects attacks and potential threats disguised in network traffic. In order to identify threats, an Intrusion Detection System known as Host-based Intrusion Detection System observes system activity through a number of log files running on the local host computer. Log files are gathered using sensors situated locally while Network Intrusion Detection Systems (NIDS) analyze the content of each packet in network traffic streams, Host-Based Intrusion Detection Systems (HIDS) rely on information stored in log files. These logs encompass Sensor recordings, system

documentation, software traces, file directories, disk provisions, user credentials and various other system logs Numerous businesses combine Network Intrusion Detection System and Host-based Intrusion Detecting System. Stateful protocol analysis, abuse identification, and identification of anomalies are used to evaluate network traffic flows. Utilizing predetermined signatures and filters, misuse detection identifies assaults. The signature database can only be kept up to date with input from humans. When it comes to identifying unknown attacks, this tactic is completely useless. It works well for identifying known assaults. Heuristic algorithms are used in anomaly detection to identify unknown hostile actions. Anomaly detection has a high rate of false positives in the majority of situations (Mishra et al. 2018). To solve this problem, the bulk of commercial systems utilize a mix of misuse and anomaly detection. This identifies deviations from acceptable protocols and applications by utilizing the established vendor standard parameters.

REVIEW OF RELATED LITERATURE

Yadav et al. (2023) proposed a hybrid intrusion detection warning system that analyze activity at the network and host levels. To adapt to and dissect curiously enormous volumes of information progressively, the framework utilized a dispersed profound learning model utilizing DNNs.

Plaka, (2021) performed thorough literature analyses on various Intrusion Detection Systems, anomaly detection methods, and machine learning algorithms suitable for detection and classification purposes.

Larsen, (2022) proposed utilizing machine learning for intrusion detection in industrial control by going through the fundamentals of ICS, such as devices communicating and their protocols. Analysis of IDS aims to gain insights into the potential implementation of AI.

Yasmeen et al. (2022) introduced a Network Intrusion Detection System employing machine learning techniques, subjected to comprehensive evaluation on performance. Multiple machine learning algorithms were assessed using the NSLKDD dataset.

Jasim and Farhan, (2022) Performed analysis of intrusion detection where a network intrusion detection system leveraging deep learning technology is proposed. The Long Short-Term Memory (LSTM) methodology was utilized in constructing a neural network employed to real data from the CSE-CIC-IDS2018 dataset for intrusion detection during data flow.

Awajan, (2023) introduced is a novel Deep Learning (DL)-based intrusion detection system tailored for IoT devices. This smart system utilizes a Fully Connected (FC) network architecture comprising four layers, designed to detect potentially harmful traffic that could initiate attacks on interconnected IoT devices.

Emad et al. (2023) introduced a system aimed at swiftly and accurately identifying threats. Employing a convolutional recurrent neural network, the study constructs a hybrid intrusion detection system based on deep learning principles, targeting network attacks using the CICIDS2018 dataset.

In view of the above, the approaches which can Successfully counter new and adversarial attacks but lack scalability to handle complex datasets and also generated data may lack meaningful features or labels, making it challenging to interpret and analyze by security analysts.

Hence, enhancing the model's performance on minority classes without compromising its ability to detect major classes is a crucial concern that requires attention. Additionally, there is a need for more up-to-date datasets encompassing diverse types of attacks to be utilized in testing and training the DenseNet Model for intrusion detection.

METHODOLOGY

DATA COLLECTION

The process of gathering data entails choosing high-quality data suitable for analysis. In this instance, we utilized the CICID2017 and CICID2018 intrusion datasets sourced from www.kaggle.com.

DATA PREPROCESSING

IJCRSSSL International Journal of Current Researches in Sciences, Social Sciences and Languages

Volume: 04 | Issue: 02 | 2024 | Open Access | Impact Factor: 5.735

The aim of preprocessing entails transforming raw data in a manner that is appropriate for deep learning. Having organized and sanitize data enables a data analyst to obtain accurate results from a deployed deep learning framework. This method includes activities like data formatting, refining, and sampling.

Model Building for DenseNet Based IDS

The model was built with an initial convolutional layer, max-pooling layer, four dense convolution (Dense Conv) blocks, and four transition layers (1x 1) Conv and 2 x 2 average pooling). Dense Convolutional blocks comprise a set of 1 x 1 and 3 x 3 Convolutional segments. After each alternative transition layer, these convolutions (1 x 1) and (3 x3) are repeated 6, 12, 48, and 32 times within each Dense Convolutional block. Resulting feature maps produced after traversing the layers act as the input for Global Average Pooling (GAP) module. Following the GAP block, there is a fully connected (FC) layer. The FC layer classifies the malware samples into their corresponding classes (Jeyaprakash et al., 2021)

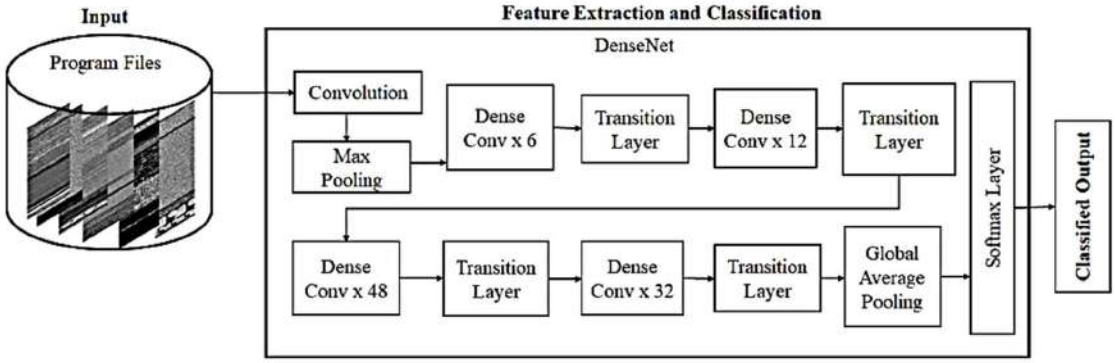

Figure 1: Flow of the DenseNet Model (Jeyaprakash et al., 2021)

The consecutive operations in the transition layer include Batch Normalization (BN), Rectified Linear Units (ReLU), and 3 × 3 convolution (Conv). Concatenation becomes impractical when the sizes of feature maps are altered. Hence, down sampling is applied to layers with different sizes of feature maps. Transition layers consist of 1 x 1 Conv and 2 x 2 mean pooling operations occur between successive Dense segments. Following the last Dense block, the classification layer is formed by connecting global average pooling with the softmax classifier. Utilizing all feature maps in the neural network, accurate predictions are made. The output layer, comprising K neurons, corresponds to the K IDS families, ensuring correct matches.

The convolution operation is responsible for learning data features while preserving connections within the feature map. Mathematically, a convolution function applies to both data and filterEach convolution layer is associated with the sequence of Batch Normalization, Rectified Linear Unit (ReLU), and Convolution. Following convolution operation on data, the Rectified Linear Unit function is utilized on the resulting feature maps. This function brings in nonlinearity into pooling in Convolutional Neural Networks is utilized to decrease the dimensionality of output feature maps, accomplished through either maximum pooling or average pooling. In maximum pooling, the highest value within defined pooling area is selected from enhanced feature map. Conversely, Average pooling partitions the input into pooling regions and calculates the mean values within each region. Global Average Pooling (GAP) calculates the average of each feature map, producing a vector that is subsequently fed into the softmax layer.

Considering that convolution is a linear operation and data often exhibit non-linear characteristics, non-linearity layers are typically positioned immediately following the convolutional layer, non-linearity is incorporated into the activation map via activation functions. These layers facilitate the learning and approximation of any continuous and complex relationships among the network's variables. Put simply, they determine the information inside the model should be activated in the forward pass and those that should not, as dictated by the network reaches its conclusion.

IJCRSSSL International Journal of Current Researches in Sciences, Social Sciences and Languages

Volume: 04 | Issue: 02 | 2024 | Open Access | Impact Factor: 5.735

There are several commonly used activation functions such as the ReLU, Softmax, Tanh and the Sigmoid functions (Salih et al., 2021) the sigmoid non-linearity is expressed mathematically as $\sigma(\kappa) = 1/(1+e^{-\kappa})$. The input to this function of activation consists of actual numbers, whereas the output is constrained within the range of zero to one in the case of the Tanh function. Similarly, resembling the sigmoid function, Tanh accepts real numbers as input but confines its output between -1 and 1., The softmax function, an alternative activation function employed in neural networks, computes a probability distribution based on a set of real numbers. This function produces an output ranging between 0 and 1, ensuring that the probabilities sum up to 1. ReLU is the most frequently utilized function in the context of CNNs. It transforms all input values into positive numbers. Lower computational load is the main benefit of ReLU over the others The ReLU function is given by ReLU(x) = max (0, x) (Salih et al., 2021).
Hence, we employed Rectified linear units and SoftMax activation function for easy optimization and for computational simplicity of our model.
Typically, connecting all features directly to the entirely connected layer can lead to overfitting on the training dataset. Overfitting happens when a model performs exceptionally well on training data but adversely affects its performance on new data. To overcome this problem, a dropout layer is utilized wherein a few neurons are dropped from the neural network during training process resulting in reduced size of the model (Salih et al., 2021).
Hence, we employed dropout after the dense layers with a value 0.5 to reduced overfitting.

DENSENET-169 ARCHITECTURE

According (Aishwarya & Padmanabha, 2023) Densenet-169, a member of the collection of DenseNet models, comprises 169 layers which is widely favored for deep learning classification tasks owing to the layered design. In contrast to alternate DenseNet structures with lesser layers, it boasts a notably reduced count of parameters that can be trained.

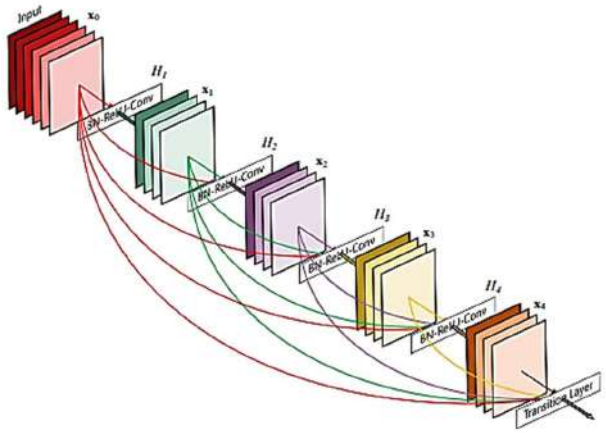

Figure 2: DenseNet-169 Architecture (Aishwarya & Padmanabha, 2023)

We can classify Densenet-169 and the other Densenet frameworks as a family of very reliable deep learning architectures due to their ability to avoid the issue of vanishing gradients have an efficient feature propagation tactics, reduce the number of trainable parameters, and promote the reuse of features. Figure 2 depicts a hierarchical structure of Densenet-169, employed in the specific study investigation. It encompasses dense layers, convolutional layers, maxpool layers, and transition layers. Throughout the model's design, the Rectified Linear Unit activation function is consistently applied, with the SoftMax activation exclusively utilized in the last layer, where convolutional layers capture features from image, while maxpool layers decrease dimensionality of the inputs. The flatten layer precedes the

densely connected layers, that operate as an artificial neural network that receive input from singular array produced by flatten layer.

Implementing Densenet-169 for an intrusion detection system involves adapting the network for a specific task of detecting intrusions in network traffic or system logs. Here is a step-by-step breakdown of how Densenet-169 can be utilized for an intrusion detection system.

Step 1: Gather the labeled network traffic or system log data for training and testing the intrusion detection system. This data should include both normal and anomalous behaviours to train the model to distinguish between the two.

Step 2: Preprocessing the collected data, which may involve task such as normalization and encoding categorical variables.

Step 3: This involves modifying the input and output layers to align with the dimension of the given input data and the number of each class representing normal and anomalous behaviours.

Step 4: Utilize transfer learning by initializing Densenet-169 with weight pre-trained on a large dataset and fine-tune the network parameter using the intrusion detection dataset to adapt it to the specific task.

Step 5: Convert the network final feature maps into a suitable format for intrusion detection. i.e., passing them through fully connected layers.

Step 6: Adjust the output layer to produce binary or multiclass prediction based on the specific requirements of the intrusion detection task.

Step 7: Adam Optimization is the selected optimizer use to train the adapted Densenet-169 for intrusion detection.

Step 8: Training the adapted model on the preprocessed and labeled dataset using Adam optimizer and regularization to prevent over-fitting.

Step 9: Evaluate the trained model using metrics suitable for intrusion detection for example accuracy, precision, recall and F1-score.

Step 10: Use Receiver Operating Characteristics (ROC) to show relationship between the parameters of the model.

Step 11: Deploy the train Densenet-169 based intrusion detection system to monitor and detect intrusion in network traffic.

By following these steps, the Densenet-169 architecture can be effectively adapted and utilized for intrusion detection task.

CONVOLUTIONAL LAYER

In simple terms, a convolutional layer applies filter to an input, producing an activation. Iterated application of filter to input generates a representation map which represents the strength of identified attributes across various locations in the input. After creating feature map, generated by multiple filters can be subjected to activation functions like ReLU. The filter size within convolutional layer is usually less than input data, and operation between these elements typically entails dot product. Considering square neuron component of size P × P succeeded by convolutional layer with filter of dimensions m × m, the resulting output from the convolutional layer would be (P − m + 1) × (P − m + 1) (Adarsh et al., 2022).

$$X_{ij}^{I} = \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} \mu_{ab} y_{(i+a)(j+b)}^{I-1} \quad (1)$$

The convolutional layer incorporates the evaluated non-linearity as depicted in Equation (2)

$$y_{ij}^{l} = \lambda\left(x_{ij}^{l}\right) \quad (2)$$

MAXPOOL LAYER

The main goal of integrating a maxpool layer in CNNs is to decrease the dimensionality of the feature map. Comparable to a convolutional layer, the maxpool layer employs a filter across the feature map, condensing the attributes within the area encompassed by the pooling filter. Suppose the feature map

IJCRSSSL International Journal of Current Researches in Sciences, Social Sciences and Languages

Volume: 04 | Issue: 02 | 2024 | Open Access | Impact Factor: 5.735

possess dimensions representing its width, height and channels, respectively. The dimensions of the feature map after implementing maximum pooling (maxp) with a filter size of f and stride of s are described by Equation (3) Adarsh et al., (2022).

$$\max_p = \frac{(n_h - f + 1)}{s} \times \frac{(n_w - f + 1)}{s} \times n_c \quad (3)$$

DENSE LAYER

According to (Adarsh et al., 2022) a dense layer in a neural network is deeply connected with its preceding layer, i.e., each neuron of the dense layer has a connection with each neuron in its preceding layer. The neuron within the dense layer aggregates input from every neuron in the previous layer and conducts a matrix-vector multiplication. Standard equation used for matrix-vector multiplication is presented in Equation (4).

$$M..\lambda = \begin{matrix} m_{11} & m_{12} \ldots\ldots\ldots\ldots & m_{1y} & p_1 \\ m_{21} & m_{22} \ldots\ldots\ldots\ldots & m_{2n} & p_2 \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots\ldots\ldots\ldots\ldots & \vdots & \vdots \\ m_{x1} & m_{x2} \ldots\ldots\ldots\ldots & m_{xy} & p_y \end{matrix} \quad (4)$$

The matrix M in the above equation represents dimensions of x × y, while matrix p is of size 1 × y. The λ matrix variable represents trained parameters of previous layer, which are subject to updates through backpropagation during training. Backpropagation modifies the weights (ωly) and biases (Bly) associated with layer ly of the neural network using Equations (5) and (6) with respect to the learning rate α.

$$w^{ly} = w^{ly} - \alpha \times dw^{ly} \quad (5)$$

$$B^{ly} = B^{ly} - \alpha \times dB^{ly} \quad (6)$$

The calculation of dω and db relies on the chain rule, which traces from the ultimate layer back to the initial layer through the hidden layers. These dω and db represent the gradient of loss function with respect to ω and b, respectively. Equations (7)–(10) are employed for computing dω and db.

$$dw^{ly} = \frac{\partial L}{\partial w^{ly}} = \frac{1}{n} dZ^{ly} A^{(ly-1)T} \quad (7)$$

$$dB^{ly} = \frac{\partial L}{\partial B^{ly}} = \sum_{i=1}^{n} dZ^{ly(i)} \quad (8)$$

$$dA^{ly-1} = \frac{\partial L}{\partial A^{ly-1}} = W^{ly^T} dZ^{ly} \quad (9)$$

$$dZ^{ly} = dA^{ly} \times g'(Z^{ly}) \quad (10)$$

The equations provided above, the variable Zly represents the linear activation at layer ly, and g'(Zly) denotes the derivative of the non-linear function with respect to Zly.

- **Transition Layer:** In CNNs, a transition layer is employed to simplify the model. A standard transition layer achieves this by employing a 1 × 1 convolutional layer to reduce the number of channels, while simultaneously halving altering the width and height of the input through the utilization of a filter with a stride of 2.
- **SoftMax Activation Function:** Is a commonly employed non-linear activation method, particularly prevalent in deep learning architectures for classification tasks. Equation (11) outlines the general structure of a non-linear activation function, where the weight is denoted by the variable w, and the bias is represented by the variable b, applied to input vector x.

$$y = f(w \times x + b) \tag{11}$$

The softmax function employed in the final layer of a convolutional neural network to compute the probabilities linked with every output class. Fundamentally, function of the softmax produces a value for each neuron in output layer. This value signifies the chance of the corresponding node serving as output. The function of the softmax, denoted as Θ, operates on the input ui, involving the exponential function of the input vector (evi) and the output vector (evo), both with m instances as specified in Equation (12).

$$\Theta(z)_x = \frac{\ell^{v_i}}{\sum_{y=1}^{m} \ell^{v_0}} \tag{12}$$

Using softmax for activation function, the loss function employed in this study is the binary cross-entropy loss function. Traditionally, binary cross-entropy is usually applied in binary classification scenarios. Equations (13) and (14) illustrate the binary cross-entropy loss function utilized in the network consisting of n layers.

$$K(W,b) = \frac{1}{n}\sum_{i=1}^{n} L\left(a^{(i)}, a^{(i)}\right) \tag{13}$$

$$L(\hat{a},a) = -(a \times \log \hat{a} + (1-a) \times \log(1-\hat{a})) \tag{14}$$

In this context, variable "a" signifies the output class 1, while "(1 − a)" represents the output class 0. The symbol "aˆ" signifies the probability of output class 1, and "(1 − aˆ)" denotes probability associated with class 0.

TRAINING THE MODEL

After the building the model, two-train data is feed to the desnseNet model. We train our model on the following hyperparameters: Batch Size refers to the quantity of samples provided to the network for training purposes.

MODEL AND DATASETS VALIDATION

Validation entails assessing the quality of your model. The Model Builder employs the trained model to predict outcomes with new test data, subsequently evaluating the accuracy of these predictions. The test data is held back to validate our model. The validation is done on the testing dataset that's 20% (unseen data), which was put aside when we had to split the original dataset. The testing dataset didn't take part in the training process at all. For each and every epoch, the loss value should be decreasing and accuracy increasing.

PERFORMANCE EVALUATION

To evaluate the model's performance, we considered accuracy, specificity, and sensitivity. These metrics rely on characteristics derived from confusion matrix, such as True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN). TP represents the count of accurately classified attacks, while FN signifies attacks that are inaccurately categorized. FN denotes the quantity of normal data erroneously classified and TN denotes accurately normal data that has been classified. The following metrics will be used: Accuracy, Recall, Precision and F1-Score

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

$$\Pr ecision = \frac{TP}{TP + FP} \quad (2)$$

$$\text{Re}\, call = \frac{TP}{TP + TN} \quad (3)$$

RESULTS AND DISCUSSIONS

Model Evaluation for CSE-CIC-DS2017 and CSE-CIC-DS2018 Datasets

Table 1 below shows a test accuracy of 97% for CSE-CIC-DS2017 Dataset, which ought to be acceptable. This implies that in 3% of instances, the classification would be incorrect. Likewise, for CSE-CIC-DS2018 Dataset which shows a higher test accuracy of 99%, 1% would not be correctly classified.

Model Evaluation	CSE-CIC-DS2017	CSE-CIC-DS2018 Dataset
Test Accuracy	0.973	0.997
Test Loss	0.006262	0.000720

Table 1: Model Evaluation for CSE-CIC-DS2017 and CSE-CIC-DS2018 Datasets

Table 2: below show that the model achieves an average of 96.9% for all metrics for CSE-CIC-DS2017 Dataset and average of 99.7% for all metrics for CSE-CIC-DS2018 Dataset. Similarly, accuracy, precision, recall, and f1 score are higher for CSE-CIC-DS2018 compared to the CSE-CIC-DS2017 datasets. This also means that our algorithm has classified an equal amount of dataset as false positive (FP), as it classified false negative (FN).

IJCRSSSL
International Journal of Current Researches in Sciences, Social Sciences and Languages
Volume: 04 | Issue: 02 | 2024 | Open Access | Impact Factor: 5.735

Performance Metrics	CSE-CIC-DS2017 Dataset	CSE-CIC-DS2018 Dataset
Accuracy	0.970	0.997
Precision	0.969	0.997
Recall	0.970	0.997
F1_score	0.968	0.997

Table 2: Shows the Performance Metrics for the two datasets

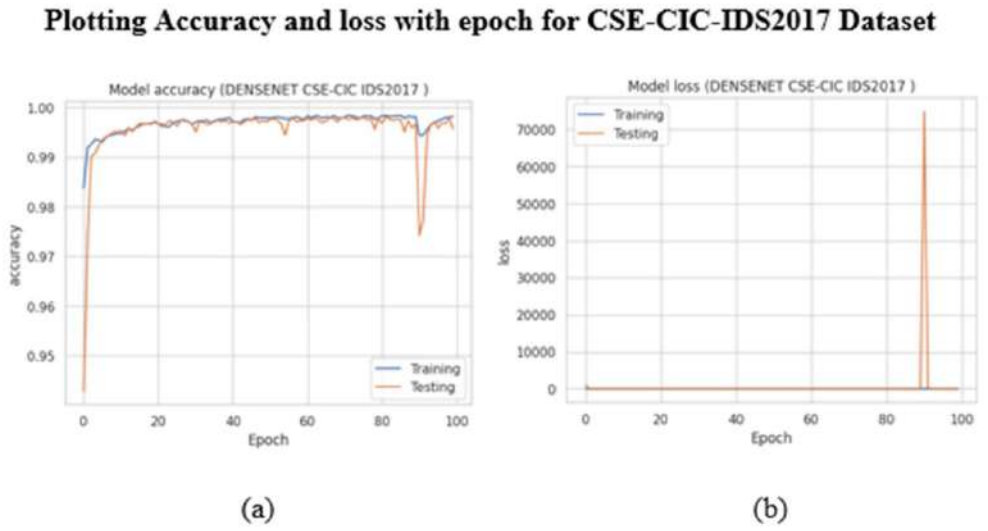

Figure 3: Accuracy and Loss graph with respect to epoch for CSE-CIC-DS2017 Dataset

Figure 3 shows the accuracy results, demonstrating that the accuracy of the proposed model consistently improves over time. At the outset, the model's accuracy appears to be low, it steadily increases over each epoch. This is attributed to the capacity for learning of the model being proposed, rendering it reliable and proficient in distinguishing between malicious and normal traffic.

CONFUSION MATRIX FOR CSE-CIC-IDS2017

It is identified that the DenseNet model wrongly-predicted-data only 16748 datasets and accurately-predicted-data 534625 out of 551373 for CSE-CIC-D 2017

IJCRSSSL
International Journal of Current Researches in Sciences, Social Sciences and Languages
International Journal of Current Researches in Sciences, Social Sciences and Languages
Volume: 04 | Issue: 02 | 2024 | Open Access | Impact Factor: 5.735

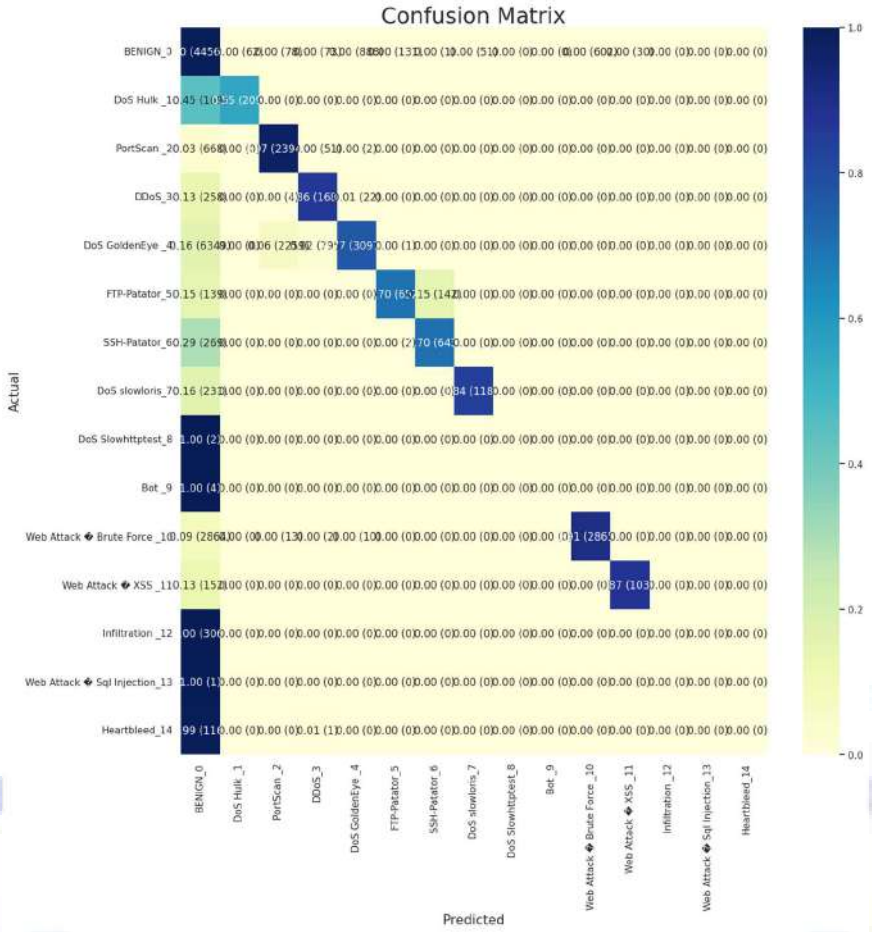

Figure 4: Confusion matrix for CSE-CIC-DS2017 dataset

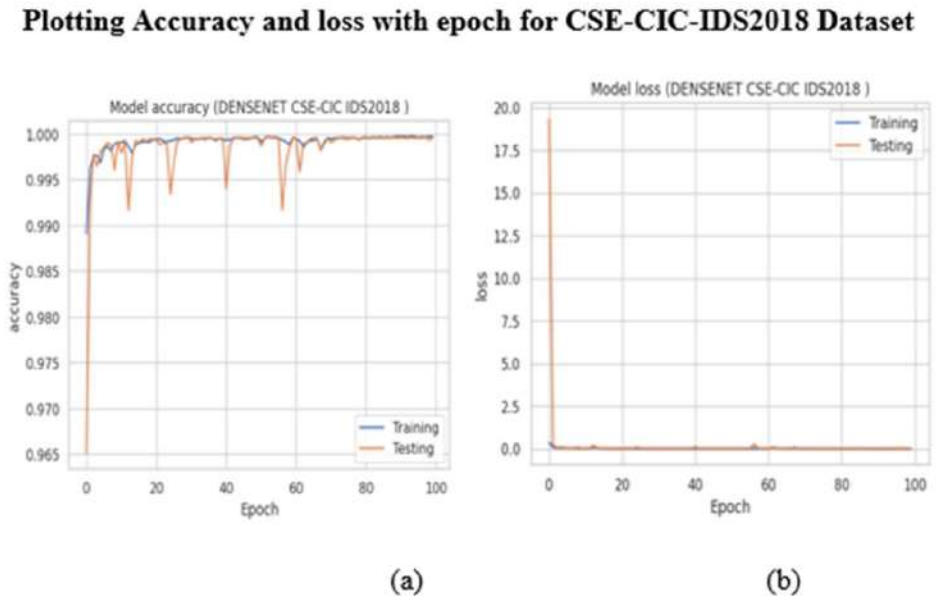

Figure 5: Accuracy and Loss graph with respect to epoch for CSE-CIC-DS2018 Dataset

IJCRSSSL
International Journal of Current Researches in Sciences, Social Sciences and Languages
International Journal of Current Researches in Sciences, Social Sciences and Languages
Volume: 04 | Issue: 02 | 2024 | Open Access | Impact Factor: 5.735

Figure 5: shows the accuracy results, demonstrating that the accuracy of the proposed model consistently improves over time. At the outset, the model's accuracy appears to be low, it steadily increases over each epoch. The outcomes for the loss graph of the proposed model demonstrate that the model's loss consistently diminishes over time. Initially, the loss graph of the model appears high, but gradually decreases with time. It is attributable to the model's learning capability which enhances its stability and proficiency in distinguishing between normal and malicious traffic.

CONFUSION MATRIX FOR CSE-CIC-IDS2018

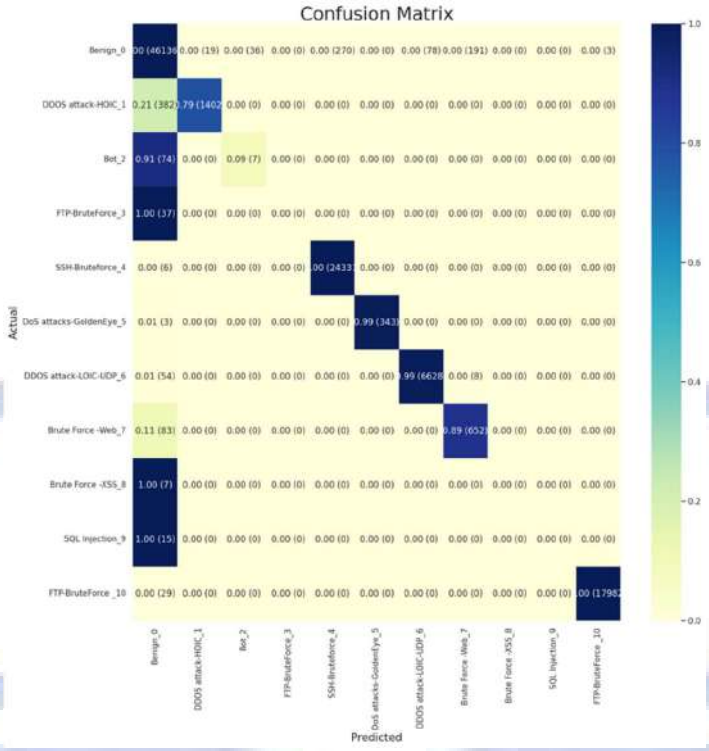

Figure 15: Confusion matrix for CSE-CIC-DS2018 dataset.

It is identified that the DenseNet model wrongly-predicted-data only 1295 dataset and accurately-predicted-data 512714 out of 514009 for CSE-CIC-DS2018.

DISCUSSION

In this work, we develop an Improved Intrusion Detection System using Densely-Connected Convolutional Neural Network to classify intrusions in the CSE-CIC-IDS2017 and CSE-CIC-IDS2018 datasets as normal or attack. In terms of overall performance on the model, CSE-CIC-IDS2018 demonstrated superior performance, achieving the highest detection accuracy of 0.997 and precision of 0.997. A high precision indicates a reliable model, contrasting with lower precision, which can lead to numerous false positives, as observed in the case of CSE-CIC-IDS2017 with an accuracy of 0.970 and precision of 0.969.

Examining the findings from both datasets, the DenseNet method outperforms the fundamental method. Specifically, overall accuracy of DenseNet method on the CSE-CIC-IDS2018 dataset is 0.997, which is 0.027 greater than that of the CSE-CIC-IDS2017 dataset. Hence, the Intrusion Detection System employing DenseNet exhibits superior performance with CSE-CIC-IDS2018 dataset compared to CSE-CIC-IDS2017 dataset.

It's important to recognize that while high recall is significant, it doesn't always signify optimal performance. Instead, a greater F1-measure indicates the model has executed exceptionally well. This is due to the F1-measure is considered the balanced average of precision and recall, providing insight into the accuracy of the model.

From the results shown in Table 2, CSE-CIC-IDS2018 had an F1-measure of 0.997 which renders it the best using the DenseNet framework.

CONCLUSION AND FUTURE RESEARCH

The proposed model primarily focus was using Densely-Connected-Convolutional Network (DenseNet) to train a network capable of defending against both recognized and unrecognized attacks. The models were trained and evaluated using two distinct datasets: CSE-CIC-DS2017 and CSE-CIC-DS2018. These datasets represent modern Intrusion Detection System datasets commonly utilized in IDS research. Training on diverse datasets enabled the model to achieve robustness, thereby enhancing its capability to counter novel attacks.

The model achieved a 96.9% using the CSE-CIC-DS2017 dataset and 99.7% for the CSE-CIC-DS2018 which are better than the other methods. The suggested detection system demonstrates high accuracy and efficiency, comparable to traditional machine learning-based solutions, while eliminating the need for manual feature engineering.

Future research calls for expanding the model incorporates additional parameters that contribute to improved performance and detection. to mitigate and detect instructions and also designing a system that not only detect but also can prevent intrusions. Finally, incorporating the model into a functioning software system or web application is a viable consideration, which would also serve to validate the model in practice.

REFERENCES

[1] Abbas, Y., Behrouz, Z., Amin, A., Ali, D., Hadis, K., Arthur, G., Conor, R., & Emely, D. (2021). A Review on Security of Smart Farming and Precision Agriculture: Security Aspects, Attacks, Threats and Countermeasures. State-of-the-Art of Cybersecurity, 11(16), 7518; doi:10.3390/app11167518.

[2] Abdulazeez, M. M., & Adnan, M. A. (2021). Neural Networks Architectures Design, and Applications: A Review. 2020 International Conference on Advanced Science and Engineering (ICOASE). DOI: 10.1109/ICOASE51841.2020.9436582.

[3] Adarsh, V., Parvathaneni, N., Madipally, S., Jana, S., Jaeyoung, C., & Muhammad, F. (2022). Fine-Tuned DenseNet-169 for Breast Cancer Metastasis Prediction Using FastAI and 1-Cycle Policy. MDPI, doi10.3390s22082988.

[4] Aishwarya, M., & Padmanabha, R. (2023). Dataset of groundnut plant leaf images for classification and detection. Elsevier Inc, doi10.1016.2023.109185.

[5] Ali, E., Juma, I., Fathi, H., & Petar, J. (2019). The Overview of Intrusion Detection System Methods and Techniques. International Scientific Conference on Information Technology and Data Related Research doi.10.15308/155-161. Sinteza: Advance computing and cloud computing .

[6] Amir, S. A., & Erik, T. (2020). A Machine Learning Approach for Intrusion Detection.University of Agder: University of Agder.

[7] Awajan, A. (2023). A Novel Deep Learning-Based Intrusion Detection System for IoT Networks. A Novel Deep Learning-Based Intrusion Detection System for IoT Networks, 12(2), 34; doi:10.3390/computers12020034.

[8] Ayyagari, N., Kesswani, M., & Kumar, K. (2021). Intrusion detection techniques in network environment: a systematic review. Retrieved from doi:10.1007/s11276-020-02529-3.

[9] Chang, G., Junkun, Y., Shenghua, Z., Pramod, K., & Hongwei, L. (2013). Long short-term memory-based deep recurrent neural. Long short-term memory-based deep recurrent neural.

[10] Chibuzor, J. U., & Bennett, E. O. (2018). An Intrusion Detection System Using Machine Learning. International Journal of Computer Science and Mathematical Theory, 6(1).

[11] Chirag, M., Dhiren, P., Bhavesh, B., Hiren, P., Avi, P., & Muttukrishnan, R. (2013). A Survey of Intrusion Detection Techniques in Cloud. Journal of Network and Computer Application, 36(1): 42-57.

[12] Clement, J. (2020). Global digital population . Retrieved from https://www.statista.com/statistics/617136/digital-population-worldwide/

[13] Dagli, Lirim A. & Cihan H., (2021). Network Intrusion Detection System using Deep Learning. Engineering Management and Systems Engineering Faculty Research and Creative works doi.10.1016,2021.05.025.

[14] Damchenas, M., Dehghantanha, A., & Mahmoud, R. (2013). A Survey on Malware Propagation, Analysis and Detection. International Journal of Cyber-security and Digital Forensics, 2(4): 10-29.

[15] Dimitri, P., Mathew, M., & Ronan, C. (2019). End-to-end acoustic modeling using convolutional neural networks for HMM-based automatic speech recognition . Lausanne, Switzerland: Speech Communication.

[16] Donatus, E. I. (2021). Network Intrusion Detection System . Finland: University of Turku.

[17] Emad, U., Muhammad, H., & Tanveer, Z. (2023). Hybrid Deep-Learning-Based Network Intrusion Detection System. Hybrid Deep-Learning-Based Network Intrusion Detection System, 13(8), 4921; Doi.10.339013084921.

[18] Farhan, R., Maolood, A. & Hassan, N. (2020). Optimized Deep Learning with Binary PSO for Intrusion Detection on CSE-CIC-IDS2018 Dataset. Journal of Al-Qadisiyah for Computer Science and Mathematics, DOI:10.29304.12.3.706.

[19] Gareth, J., Daniela, W., & Trevor, H. (2013). An Introduction to Machine Algorithm. New York Heidelberg Dordrecht London: eBook.

[20] Hindy, H. (2021). Intrusion Detection Systems using Machine Learning and Deep Learning Techniques. Abertay University.

[21] Hsiao-Chi, L., Zong-Yue, D., & Hsin-Han, C. (2020). Lightweight and Resource-Constrained Learning Network for Face Recognition with Performance Optimization. Lightweight and Resource-Constrained Learning Network for Face Recognition with Performance Optimization. 20(21), 6114; Doi:10.3390/s20216114.

[22] Ian, G., Yoshua, B., & Aaron, C. (2016). Deep Learning. Cambridge: An MIT Press book.

[23] Imran, H., Muhammad, Z., & Arshad, A. (2022). Machine Learning-Based Intrusion Detection System: An Experimental Comparison. Journal of Computational and Cognitive Engineering, 1–10 DOI: 10.47852,2202270.

[24] Jan, L., Sadib, A., Moktar, M., Mohammed, M., Sarkhel, H., Taher, K., Shima, R., Mehdi, H., & Amir, M., (2021). Deep Learning-Based Intrusion Detection. Sulaymaniyah, Iraq: Digital Object Identifier 10.1109/ACCESS.2021.3097247.

[25] Jasim, D. A., & Farhan, B.I. (2022). Performance analysis of intrusion detection for deep learning . Indonesian Journal of Electrical Engineering and Computer Science, DOI: 10.1159126.2.1165-1172.

[26] Jeyaprakash, H., Abijah, R., Subbiah, G., Seifedine, K., & Robertas, D.(2021).An Efficient DenseNet-Based Deep Learning Model for Malware Detection. Entropy Journal. DOI.10.339023030344.

[27] Jin, K., Nara, S., Seung, Y. & Sang, H. K. (2017). Method of Intrusion Detection using Deep Neural. Seoul, Korea: Seculayer Co., Ltd.

[28] John, R. A., Jyotibdha, A., Chao, Z.A., Surendran, S.K., Bose, A.C., Nidhi, T.Y., Gao, Y. H., Keke, K. Z., Manzhang, X. W., Lin, L. Z., Liu, A. B. & Nripan, M. (2020). Optogenetics inspired transition metal dichalcogenide neuristors for in-memory deep recurrent neural networks. Nature Communications. Journal of Electrical and Electronics Engineering, DOI: 10.103841467-020- 16985-0.

[29] Katleho, M., Ali, H., & Thokozani, S. (2020). Deep Learning in Object Detection: a Review. International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD) DOI:10.110949160.2020.9183866.

[30] Laith, A., Jinglan, Z., Amjah, J., Ayad, A., Ye, D., Omran, A., Santamaria, J., Mohammed, A., Mithana, A., & Laith F., (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of Big Data, 8(53).

[31] Lang, H., & Lui, B. (2020). Multidisciplinary Digital Publishing Institute. Retrieved from machine learning and deep learning method for intrusion detection system: Doi:2076-3417/9/20/4396

[32] Larsen, J. N. (2022). Using Machine Learning to Detect Intrusions in Industrial Control Systems. Oslo, Norway: University of Oslo.

[33] Latha, S., & Prakash, S. J. (2017). A Survey on Network Attacks and Intrusion Detection System in Advanced Computing and Communication System (ICACCS). Fourth International Conference on The Institute of Electrical and Electronics Engineers (IEEE) DOI: 10.1109/ICACCS.2017.8014614).

[34] Mahesh, B. (2020). Machine Learning Algorithms - A Review. International Journal of Science and Research (IJSR), 7.426.

[35] Mishra, P., Varadharajan, V., Tupakala, U., & Pillin, E. S. (2018). A Detailed Investigation and Analysis of using Machine Learning Technique for Intrusion Detection. journal of Institute Of Electrical and Electronics Engineers, (99):DIO: 10.1109,2018.2847722.

[36] Mohssen, M., & Al-Sakib, K. P., (2013). Automatic Defense Against Zero-day Polymorphic Worms in Communication Networks. doi.10.1201,14912,337.

[37] Nabe, C. (2021). Impact of COVID-19 on cybersecurity. Retrieved from https://www2.deloitte.com/ch/en/pages/risk/articles/impact-covidcybersecurity.html.

[38] Nasira, D. S. (2016). Detecting and Preventing Intruion in Multi-tier Web Application using Double Guard . Third International Conference on Computing for Sustainable Global Develoment (INSPEC Accession Number: 16426087). New Delhi, India: IEEE.

[39] Noor, W., Mohammed, S., Alshehri, M., Sultan, A., Naghmeh, M., Abdulwahab, A., Safi, U., Naila, N., & Jawab, A. (2023). A hybrid deep learning-based intrusion detection system for IoT networks. Edinburgh, UK: Mathematics and Biosciences Engineering.

[40] Olatunde, I., Festus, A., & Alaba, T.O. (2019). Intrusion Detection using Deep Learning Technique: A Review. Elizade University, Ilara-Mokin, Nigeria: Annals. Computer Science Series. 17.

[41] Orcid, Ö., Batur, D., & Nizamettin, A. (2020). An Optimal Feature Parameter Set Based on Gated Recurrent Unit Recurrent Neural Networks for Speech Segment Detection. An Optimal Feature Parameter Set Based on Gated Recurrent Unit Recurrent Neural Networks for Speech Segment Detection. 10(4), 1273; Doi:10.3390,10041273.

[42] Ping, K., Tingsong, M., Ziwei, C., & Fan, L. (2018). Image super-resolution with densely connected convolutional networks. Image super-resolution with densely connected convolutional networks. DOI:10.1007,10489-018-1234.

[43] Plaka, R. (2021). Instrusion Detection using Machine Learning for Industrial Control System. Västerås, Sweden: Mälardalen University.

[44] Przemysław, B., & Bartosz, J. (2015). An Entropy-Based Network Anomaly Detection Method. An Entropy-Based Network Anomaly Detection Method, 17(4), 2367-2408; Doi:10.3390,17042367.

[45] Rahman, T. (2022). Intrusion Detection system based. Norway: Aalto University.

[46] Richard, S., Alex, P., Jean, W., Jason, C., Christopher, D.M., Andrew, N., & Christpher, P. (2013). Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. Conference on Empirical Methods in Natural Language Processing (1631–1642). Seattle, Washington, USA: Association for Computational Linguistic.

[47] Rohini, A. N., Pooja, D. R., & Abhay, D. R. (2016). A Review of Intrusion Detection System Basic Concepts. International Journal of Computer Science and Mobile Computing, 5,482-485.

[48] Salih, A. A., Ameen, S. Y., Zeebaree, S. R. M., Sadeeq, M. A. M., Kak, S. F., Omar, N., Ibrahim, I. M., Yasin, H. M., Rashid, Z. N., & Ageed, Z. S. (2021). Deep Learning Approaches for Intrusion Detection. 9(4), 50–64. https://doi.org/10.9734/AJRCOS/2021/v9i430229

[49] Saravanan, M. S., & Dhikhi, T. (2019). An Enhanced Intelligent Intrusion Detection.International Journal of Innovative Technology and Exploring Engineering (IJITEE), 8.

[50] Shah, S., & Pramila, P. (2019). A Review of Machine Learning and Deep Learning Applications. Fourth International Conference on Computing Communication Control and Automation (ICCUBEA) DOI: 10.1109,8697857.

[51] Taye, M. M. (2023). Understanding of Machine Learning with Deep Learning:. journal of message passing interface, Doi:10.3390,12050091

[52] Weicong, K., Zhao, Y., Youwei, J., David, J. H., Yan, X., & Yuan, Z. (2017). Short-Term Residential Load Forecasting based on LSTM Recurrent Neural Network. Short-Term Residential Load Forecasting based on LSTM Recurrent Neural Network, DOI: 10.1109,2753802.

[53] Weili, F., Peter, E. D., Hanbin, L., & Lieyun, D. (2020). Computer vision for behaviour-based safety in construction: A review and future directions. Columbia University, New York 10027, USA.

[54] Yadav, Prateek S., & Rajesh. (2023). Deep Learning Approach for Intelligent Intrusion Detection System. Proceedings of the International Conference on Innovative Computing & Communication (ICICC) (ssrn.com/abstract=4386519).

[55] Yanfang, F., Yishuai, D., Zijian, C., Qiang, L., & Wei, X. (2022). A Deep Learning Model for Network Intrusion Detection with Imbalanced Data. A Deep Learning Model for Network Intrusion Detection with Imbalanced Data, : 11(6), 898; Doi:10.3390,11060898.

[56] Yasmeen, S.A., Bader, A., & Munshi, A. A. (2022). Network Intrusion Detection Using Machine Learning Techniques. Advances in Science and Technology, Doi:10.12913/22998624/149934.

