OPTIMIZING INTRUSION DETECTION PERFORMANCE THROUGH MLP NEURAL NETWORKS

Sabeena S[*1], Dr. Chitra S[2]

[*1]Research Scholar, Department of Computer Science, Bishop Heber College, Bharathidasan University, Trichy, Tamil Nadu, India

[2]Assistant professor, Department of Computer Science, Bishop Heber College, Bharathidasan University, Trichy, Tamil Nadu, India

[*1]Corresponding author Mail Id: sabeenashameem1990@gmail.com

Abstract

In the technological environment, cyber-attacks are popular and becoming more sophisticated, and existing intrusion detection models are not adequate in hard threat escapes. The reason for increasing demands in the network systems, malicious attackers are increasing their involvement in these areas. Numerous organisations or network-based environments have been tackling hundreds of attacks on the network frequently. Conventionally, various manual methods have been used for intrusion detection, like traffic log reviews, flow and packet analysis and security monitoring. Nonetheless, these techniques are found to be time-consuming and inefficient for obtaining better results, and manual methods are unable to detect all types of attacks for network security. To overcome these problems, several researchers have focused on intrusion detection systems to provide effective security to the networks. Conversely, it results in speed and accuracy lacks. For improving the intrusion detection, the proposed research employs a Deep Learning (DL) approach, an Encoded Multi-Layer Perceptron (Encoded MLP) with an attention mechanism. For the process of feature optimisation, the proposed model uses the Adaptive Beetle Algorithm, which is used to provide accurate results for the detection model. It uses the RT-IoT dataset for evaluation. Correspondingly, the efficacy of the projected model is calculated utilising various evaluation metrics like F1 Score, recall, accuracy, and precision to estimate the performance of the proposed DL research. Furthermore, the comparison of existing models discloses the efficiency of the proposed method. The present research is envisioned to contribute to the emerging intrusion detection methods for network and cybersecurity against hackers and malicious attacks.

Keywords: Intrusion Detection, Intrusion Detection System (IDS), Auto Encoder Multi-Layer Perceptron, Adaptive Beetle Algorithm, Attention Mechanism and Deep Learning (DL).

1. Introduction

Globally, computer networks and the internet have revolutionised the aspect of communication with each other, enabling data sharing at an unprecedented rate. However, the pervasive interconnection has created various opportunities for malicious actors to abuse the vulnerabilities and unauthorised access to several sensitive information [1]. SDN brings many benefits like flexibility, programmability and manageability [2]. An effective intrusion detection model is required to secure the network infrastructure from various types of cyberattacks. Moreover, an intrusion detection system is a security tool utilised to monitor the network activity and protect authenticated users from malicious activities that threaten dependability, accessibility and security of sensitive information [3]. In accordance with network security, Software Defined Networks architecture has its unique security requirements and several lacks in security flaws make the architecture vulnerable to various malicious threats and attacks. Recently, researchers have utilised Artificial Intelligence (AI) based technology for Intrusion Detection Systems. Hence, the advantages of Machine Learning (ML) and DL offer various applications in Network security. The implementation of DL in a network-based environment is a notable aspect of the platform.

Contrastingly, the conventional model has employed a data-plane ML solution. The ML solution has been evaluated with the help of RF, SVM and KNN to detect DDoS attacks in real network traces and the results have shown that the Data plane-ML is faster than the statistical methods [4, 5]. Similarly, the existing model has presented a combination of the SMOTE over-sampling method and 1-D CNN method [6]for minority classes of attack classification. It has utilised the UNSW-NB15 dataset for the testing process. The outcomes showed that the prevailing model has attained better values on evaluation metrics [7]. Correspondingly, a conventional model has developed anomaly anomaly-based intrusion prediction system for the IoT networks. Later, it has

implemented utilising CNN in 1D, 2D and 3D. It has been validated on four datasets, namely IoT Network Intrusion, BoT-IoT, IoT-23 intrusion detection and MQTT-IoT-IDS2020 dataset [8], which have attained better values in terms of accuracy [9]. Accordingly, the conventional models accomplished satisfactory results. However, it lacks limitations such as accuracy, over-fitting of data, computational cost and low detection performances.

To solve these issues, the proposed model uses a certain set of procedures to improve the performance in detection through classification of intrusions and attacks in a network-based environment. The proposed model utilises a publicly available dataset, namely RT-IoT. Once the input data are loaded, it processes the pre-processing technique to perform missing values and feature optimisation is processed with the help of the Adaptive Beetle Algorithm. Then, pre-processed data are divided into two in the ratio 80:20, where 80 per cent is for the training process and 20 per cent for the testing process. The training set is passed to the proposed DL model of Auto-encoder MLP with attention mechanism to perform the intrusion classification. Then, the testing data is tested by the detection model and evaluated through performance metrics. The major contributions of the present DL model are provided below.

- To utilise a DL model for intrusion detection with the RT-IoT dataset to enhance the computation and accuracy in the present research.
- To employ Auto auto-encoder MLP with an attention mechanism to perform anomaly classification for network security.
- To calculate the efficiency of the present model with performance metrics like F1 score, accuracy, precision, recall and ROC curve.

1.1. Paper Organization

The flow of the present model is given here: Section 1 provides an overview of the background of the proposed model. Section 2 reviews the conventional literature related to intrusion detection and problem identification. Section 3 precisely describes about proposed methodology. Furthermore, section 4 provides a table and graphical representation of data analysis. Section 5 discusses the results of the proposed model with traditional research. Finally, section 6 concludes the present model along with future research.

2. Literature Review

This section provides an analysis of various existing research on intrusion detection, along with other techniques for the prediction of intrusion classification systems.

For instance, the conventional model has developed an adaptive and robust Intrusion Detection System (IDS) to predict and classify network attacks. It has utilised the UNSW-NB15 dataset, which has obtained 95.4% of accuracy in pre-partitioned classification and 95.6% of accuracy in user-defined multiclass classification [10]. Similarly, the prevailing model has developed some ML and DL strategies to improve the Internet of Things (IoT) security performance. It has utilised an existing dataset, namely NSL-KDD and BoT- IoT. The existing CNN has attained 90.85% of mean accuracy in 10 epochs at a batch size of 32, whereas MLP has attained 54.10% of mean accuracy in 10 epochs at a batch size of 128 [11]. Contrarily, the traditional model employed DDoS mitigation in IoT-based environments with fog computing and ML techniques. It has used a customised network dataset, and the results have shown 77% accuracy [12]. Correspondingly, a CNN-based technique has been deployed for anomaly intrusion detection systems (IDS), which took IoT's advantage. It has provided the ability to analyse the whole traffic across IoT. It has utilised the NID and BoT-IoT dataset and has attained better results [13]. Likewise, a deep CNN model has been deployed for classifying and detecting nearest real-time network intrusions from the imbalanced cloud surrounding. In addition, the evaluation has been carried out with the CSE-CIC-IDS2018 dataset that has attained 97.07% of testing accuracy [14].

Security is one of the major and vital parameters for a Wireless Sensor network. Hence, security could be developed through identifying the DoS attacks [15]. Thus, the suggested model has employed 2 kinds of ML classifiers, which involved Neural Network (NN) and Support Vector Machine (SVM) for detecting malicious attacks in the network systems. NN, a distributed parallel method and a global minimum that has been found by training the SVM using a kernel. In accordance with experimental results, it has been identified that SVM has attained better accuracy than NN [16]. In a similar vein, the traditional approach has employed different ML classification algorithms such as Naïve Bayes (NB), Decision Tree (DT), Random Forest (RF), Decision Table (DTb), Random Decision Forest (RDT), and Artificial Neural Network (ANN) to predict intrusions to offer services in the cybersecurity domain. It has utilised a cybersecurity dataset. 93%, 94%, 90%, 92%, 91%, 91% and

90% of accuracy have been attained by DT, RF, RT, DTb, ANN, NB and BN, respectively [17]. Additionally, the existing model has deployed a DNN technique for anomaly detection in the Network Intrusion detection mechanism. It has utilised the NSL_KDD dataset, and a softmax layer has been used along with a cross-entropy loss mechanism for enforcing the conventional model in various classifications comprising 5 labels like normal, DoS, R2L, Probe and U2L. It has attained better results [18]. Similarly, the traditional model has developed a Difficult Set Sampling Technique (DSSTE) for intrusion prediction in unjust network traffic. It has utilised NSL-KDD and CSE-CIC-IDS2018 datasets. The existing DSSTE with RF, SVM, XG Boost, LSTM, AlexNet and mini VGGNet have attained 80.5%, 77.5%, 80.13%, 81.78%, 82.84% and 81.27% of accuracy, respectively [19].

Contrastingly, the prevailing CNN-LSTM has been employed for extracting the real HTTP traffic characteristics without compression, calculating entropy and encryption. It has used two datasets, namely CICIDS2017and CSIC-2010, which have attained better accuracy [20]. Similarly, the conventional model employed fast entropy-based DDoS detection. It has analysed CNN and Stacked Auto Encoder (SAE), which have attained 93% and 94% of accuracy, respectively [21, 22]. Correspondingly, the existing model has deployed a Generative Adversarial Network (GAN), a DL approach framework for the detection of DDoS attacks. It utilised the public CICDDoS 2019 dataset. The results have shown that the LSTM, GAN and MLP have attained 90.29%, 94% and 92.12% accuracy, respectively [23]. The existing model has proposed to categorise benign traffic from DDoS attacks through using ML techniques, and has also identified attack detections. It has created an SDN dataset, and the outcomes have shown that hybrid SVC-RF has attained 98.8% of accuracy with a low false alarm rate [24]. Moreover, the conventional research has employed an IDS framework utilising ML methods. It used a Recurrent Neural Network (RNN), LSTM and Gated Recurrent Unit (GRU). It has utilised UNSW-NB15 and NSL-KDD datasets. Therefore, XG Boost-LSTM has attained 88.13% of test accuracy in binary classification. For the multi-class classification, prevailing XG Boost –LSTM has attained 86.93% of test accuracy on the NSL-KDD dataset [25].

2.1. Problem Identification

- Though the conventional model has produced satisfactory results, it needs improvements in feature extraction or optimising techniques [10].
- Accuracy is a significant evaluation metric utilised to analyse the model's performance. Nevertheless, classical research has lacked in the accuracy value [10] [11] [12].

3. Proposed Methodology

The proposed method identifies and extracts information from the given dataset for intrusion detection. The detection is carried out by implementing auto auto-encoder MLP, a DL model with an attention mechanism. Existing works on intrusion detection have produced inaccurate results with slow speed convergence. Therefore, the proposed model utilised the DL algorithm for the detection of RT-IoT data. Moreover, the flow of the proposed DL method is depicted in the figure below. 1.

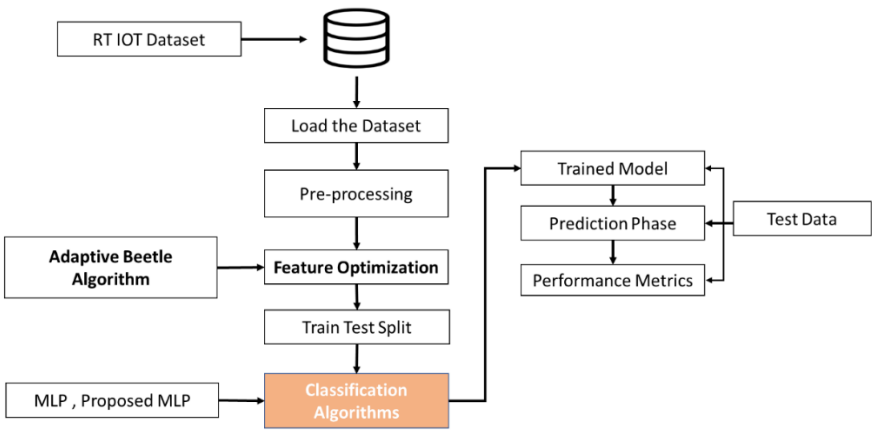

Figure 1 Design of the Proposed Method

The figure. 1 deliberates on the proposed model's flow. It signifies the proposed system creation for anomaly classification in a network-based environment. It comprises dataset loading, data pre-processing, and feature optimisation with the help of the Adaptive Beetle algorithm, data splitting and classification process using MLP and the proposed MLP model. The following figure. Figure 2 shows the mechanism of the proposed method.

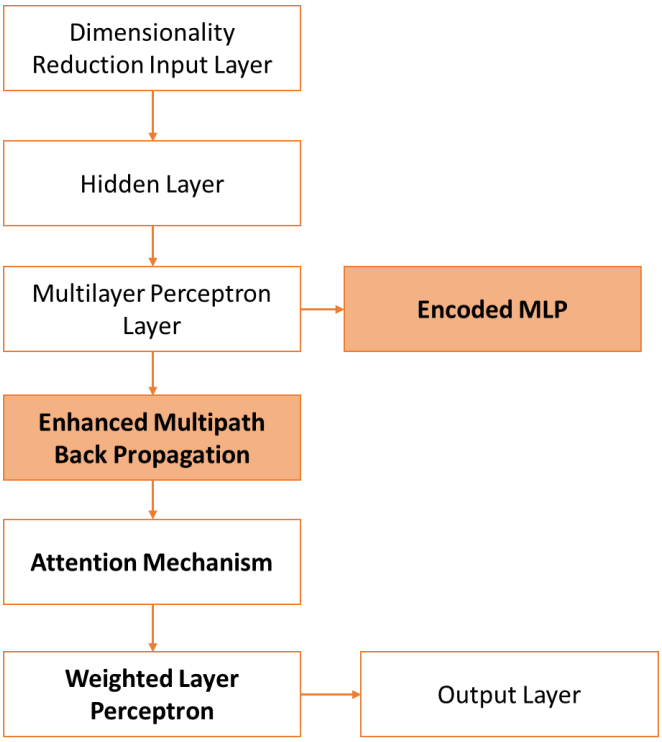

Figure 2: Mechanism of the Proposed Method

Figure 2 shows the mechanism of the proposed model. It shows that the MLP layer has been incorporated with the encoded MLP. It is further processed with enhanced multipath backpropagation with an attention mechanism. Then, the data are processed with a weighted layer perceptron to get the optimal output.

3.1. Dataset Description

The present method utilises the RT-IoT dataset, which is derived from real real-time IoT infrastructure. RT-IoT dataset embraces both adversarial and normal network behaviours. It incorporates data from IoT devices like Wipro-Bulb, simulated attack scenarios, MQTT-temp and ThingSpeak LED. This dataset provides a precise perspective on the complicated nature of the network traffic. Hence, the proposed model uses the RT-IoT dataset to provide advanced capabilities of IDS and provide security for IoT networks. The official link to the utilised dataset is provided below**.** Dataset link: **https://archive.ics.uci.edu/dataset/942/rt-iot2022**

3.2. Data Pre-processing

Data Pre-processing is a method of changing the raw data into a proper data set, which is processed to check the missing values, noisy signals, label encoding, feature scaling and other inconsistencies, before being applied to the algorithm. Besides, the pre-processing technique improves the detection performance of the proposed method. To achieve this, the proposed model executed two significant pre-processing techniques, like checking missing values and label encoding. In the process of label encoding, the categorical data values are assigned to numerical labels to enhance the efficiency of the proposed model.

3.3. Feature Optimisation-Adaptive Beetle Algorithm

The present model used feature optimisation for selecting the useful and most relevant data from the dataset to enhance the performance of the DL model. It involves recognising and choosing the significant features that could

contribute to the proposed detection model while reducing redundancy and noise. These methods are aimed at improving the accuracy, interpretability and efficiency through focusing on relevant input data. For the feature optimisation process, the proposed method utilised the Adaptive Beetle Algorithm, which has a beetle group behaviour. Furthermore, the Beetle optimisation algorithm has been widely utilised for various optimisation problems, which has better convergence speed and provides accurate results.

3.4. Data Splitting

In DL, data splitting is used to eradicate data overfitting. Basically, DL uses the data splitting technique to train the respective model, where the training data is added to the proposed method to equip the training stage parameters. After the training process, the test set data are measured to calculate the present model for handling the observations. In the present model, the original data is split into 2 sets in the ratio 80:20, which indicates that eighty per cent of the new observations are utilised for the training, and the residual 20 per cent of the observations are applied for testing to calculate the performance of the respective technique.

3.5. Classification - Encoded MLP: Connectivity with Weighted MLP and Enhanced Multipath Back Propagation

3.5.1 Conventional MLP Model

ANN feeds the information in the forward direction. It consists of one input layer and one output layer, one or more hidden layers. Each layer is made up of one or more neurons, which are used to apply the activation function to the given input. Neurons add a value called bias, in which each link among nodes has a corresponding weight to its network importance. MLP has more hidden layers and is mainly utilised for problem-solving tasks that need supervised learning. In MLP, the computations are challenging and time-consuming. It is difficult to interpret the prediction model. It is essential to pre-process the data and enhance the features to obtain optimal performance. Additionally, MLPs are often employed for intrusion detection models. Figure. Figure 3 illustrates the architecture of a conventional MLP model.

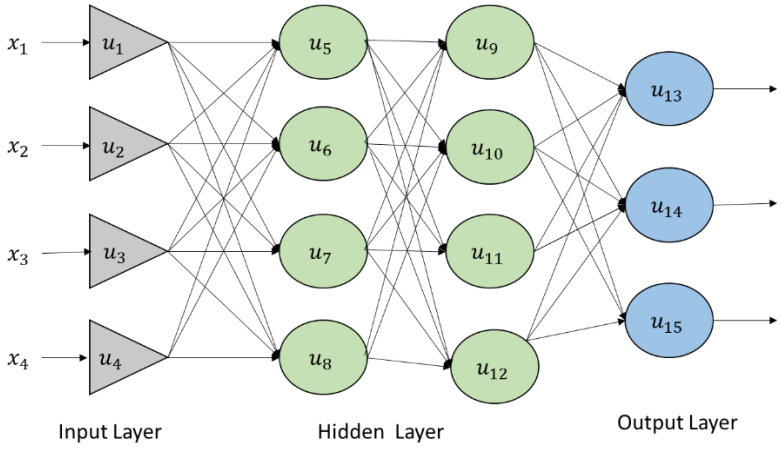

Figure. 3 Architecture of Conventional MLP Model

The figure. Figure 3 depicts the architecture of a conventional MLP model. The architecture comprises of input layer, hidden layers and output layers. The input layers are named as u1, u2, u3 and u4. The hidden layers are from u5 to u12, whereas the output layers are from u13 to u15. Moreover, MLP is trained through deploying a static backpropagation learning rule where the MLP has many layers that increase the network's learning ability. The non-linear activations are executed to leverage high non-linear tasks. Input signals are processed in all layers in the network, while the output is computed in the output layer. Consequently, error gradients are computed and backpropagated to the network by each layer. The pseudo code for the MLP is provided below, which describes the training algorithm of the MLP network.

Pseudo code for Multi-Layer Perceptron

Step 1: Initialise the MLP network parameters, weights and bias

Step 2 : $X_i = (x_1, x_2 \dots x_r)$

Step 3: $p_{in\ j} = \sum_{j=1}^{r} X_i\, c_{ij}$ // input layer

$p_j = f(\sum_{j=1}^{r} X_i\, c_{ij})$ // hidden layer

$Y_{irk} = \sum_{j=1}^{r} p_i\, l_{ij}$ // output layer

$Y_k = f(\sum_{j=1}^{r} p_i\, l_{ij}))$

Step 4: *The error between actual and target variable*

Step 5: Error back − propagation learning algorithm
is implemented to update the weight vectors

Step 6: Check for stopping criteria, in the proposed study
The error convergence to the rate of $10^{-5} is$, the stopping criteri
Once the stopping criteria are attained
Stop the algorithm and return the solution

3.5.2 Proposed Auto Encoder MLP

The proposed method employs a DL approach to detect network intrusions effectively using an Auto Encoder and an MLP. The proposed model can attain higher efficiency in network data classification into normal and attack with better accuracy, false alarm rates and high detection rates. The basic objectives of the auto encoders are known to reconstruct input data, rebuilding which can decrease the original features related to the number of features specified. Hence, the proposed model passes the raw data to the encoder. Next, the training and testing set of data are implemented to detect the intrusion using the phase of the encoder. The architecture of the Autoencoder MLP is provided in the figure. 4.

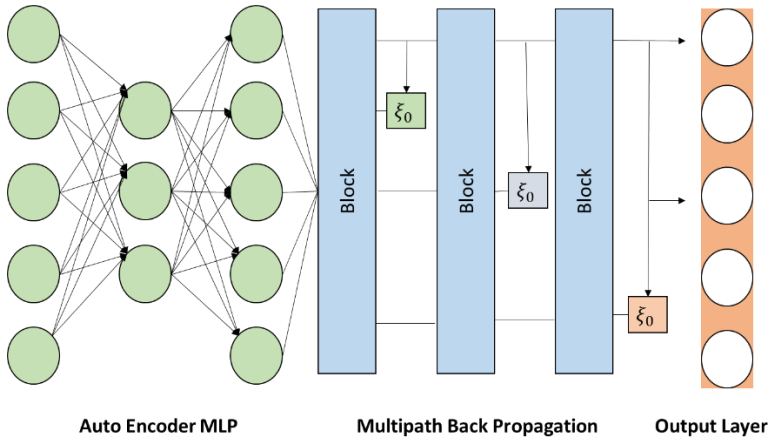

Figure 4: Architecture of Auto Encoder MLP

Auto encoder is an ANN that teaches an encoded version of the raw data or input data. The encoder takes raw data and derives its features, where the decoder utilises the characteristics for data reconstruction. Auto encoder constructions are included with an input tier, a latent tier and an output tier. Moreover, the input and output tiers are equal in size, whereas the latent tier is smaller than the input tier. For the provided training set data $v = \{, , \dots, v_q\}$ along with m samples, v_i is a vector of d-dimensional attributes, encoder traces input vector v_i to hidden representation vector h_i by mapping f_θ in equation 1,

$$h_i = f_\theta(v_i) = s(c_{vi} + b) \quad (1)$$

Here, $c,\ \acute{d} \times d, \acute{d}$ refers to hidden units, θ is the mapping parameter set $\theta = \{c, b\}$, and b is the bias vector. Moreover, the sigmoid function $s\ is\ is\ is$ denoted in equation (2).

$$s(t) = \frac{1}{1+exp-t} \quad (2)$$

Here t is a parameter that determines the form of the function. Decoder converts the return in final hidden representation $h_i$to rebuild they_i, d-dimensional vector, as in input space as provided in equation (3).

$$y_i = g_\theta(v_i) = s(c_{hi} + b) \quad (3)$$

where $\acute{c}$ is $d \times \acute{d}$, $\acute{\theta} = \{\{\acute{c}, \acute{b}\}\}$ and $\acute{b}$, a bias vector. Auto autoencoder is trained to decrease the variance between output and input. Consequently, equation (4) I to measure the loss function.

$$\mathcal{L}(\mathrm{v},\mathrm{y}) = {}^{1}/_{m} \sum m_i = 1 \,||\, v_i - y_i \,||2 \quad (4)$$

where m is the number of the training dataset. Finding the better parameters is a target(θ and. It efficiently decreases the disparity between input and rebuilds the output among the whole training set, as in equation (5).

$$\theta = \{\mathrm{c}, \mathrm{b}\} = arg_\theta minL(v, y) \quad (5)$$

The pseudo code for the Encoded MLP is provided below.

Pseudo code for Encoded MLP

Model parameters: $\{c_l\}_{l=0}^{p-1} = \{c_0, \ldots, c_{L-1}\}.Y_c$.θ ;
Output positions: $\{L_i\}_{i=0}^{c} = \{L_0, \ldots, L_c\}$;
Weights for different losses $\gamma i_{i=0}^{c} = 0 = \{\gamma 0, \ldots, \gamma c\}$;
Input data: x0; Learning rate: α.
Shared Forward Propagation for multiple outputs
For l = 0 to p − 1 do
Compute $X_{l+1} =$ h $(\lambda_l v_l + F_l(v_l, c_l))$;
if $l \in \{L_i\}_{i=0}^{c}$ *then*
Compute the loss ξ_i;
end if
end for
Multi − way Backwards Propagation
Let $U_l^0 <- c_l, \forall l \in \{0, \ldots L-1\}$;
for i = 0 to w do
Conduct backpropagation w. r. t. ξi to compute $\left\{\frac{\partial \xi i}{\partial U_l^i}\right\}_{l=0}^{L_i}$.
Update model parameters that are relevant to ξi by
$U_l^{i+1} <- U_l^i - \propto \gamma i, \forall l \in \{0, \ldots L_i\}$
Update model parameters that are irrelevant to ξi by
$U_l^{i+1} <- U_l^i, \forall l \in \{L_i + 1, \ldots L-1\}$
end for
Let $c_l <- U_l^{K+1}, \forall l \in \{L_i + 1, \ldots L-1\}$

Table 1 provides a detailed comparison of the key components and hyper parameters used in the Convolutional Neural Network (CNN) and Multi-Layer Perceptron (MLP) models. It summarises the architecture, loss functions, optimisation techniques, and hyper parameter tuning strategies for both models, offering a clear overview of the experimental setup.

Table 1: Comparison of Model Architectures, Loss Functions, and Hyper parameter Tuning for CNN and Proposed MLP

Aspect	Convolutional Neural Network	Proposed Multi-Layer Perceptron
Architecture	Convolutional layers with ReLU activations, max-pooling, and fully connected layers	Fully connected layers with ReLU activations

Activation Functions	ReLU (hidden layers), Softmax (output layer)	ReLU (hidden layers), Softmax (output layer)
Loss Function	Categorical Cross-Entropy	Categorical Cross-Entropy
Optimizer	Adam Optimizer	Adam Optimizer
Learning Rate	Range tested: 0.001 – 0.1 (with learning rate scheduler)	Range tested: 0.001 – 0.1 (with learning rate scheduler)
Number of Layers	Varies (Convolutional + Fully connected layers)	Varies (Fully connected layers only)
Neurons per Layer	Varies depending on the architecture	Varies depending on the architecture
Batch Size	Tested: 32, 64, 128	Tested: 32, 64, 128
Epochs	Up to 50 epochs	Up to 50 epochs
Hyperparameter Tuning	Grid search and cross-validation	Grid search and cross-validation

Thus, the proposed Autoencoder-MLP model and a conventional MLP lies in the incorporation of the Autoencoder for feature extraction. While a traditional MLP directly processes the raw input data through multiple hidden layers to make predictions, the proposed method first employs an Autoencoder to reconstruct and reduce the dimensionality of the input features. This preprocessing step helps to enhance the quality of the input data, improving the efficiency of the MLP in detecting network intrusions. The Autoencoder effectively reduces noise and irrelevant features, leading to better accuracy, lower false alarm rates, and higher detection rates in the intrusion detection task.

4. Results and Discussion

This division represents and analyses the outcomes of the proposed method that detects the intrusions in network systems to provide network security with Auto auto-encoder MLP with back propagation and an attention mechanism.

4.1. Exploratory Data Analysis

This section provides EDA of the RT-IoT dataset in the respective model. The EDA is applied to visualise and understand the data in the generated dataset. Figure 5 and Figure. Figure 6 shows attack counts and attack types in the dataset.

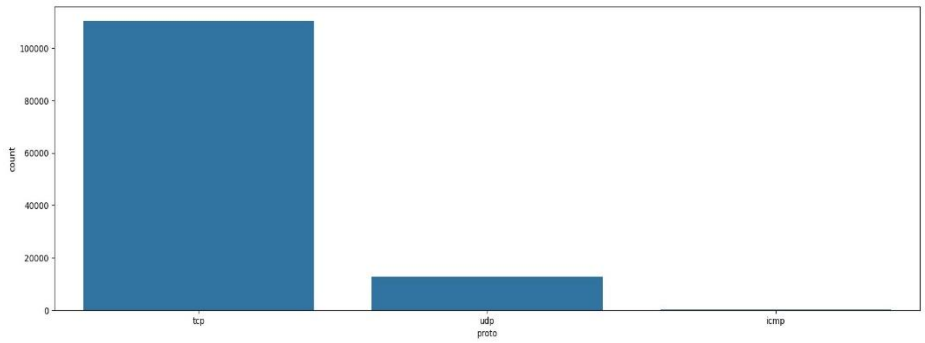

Figure 5 Attack Counts

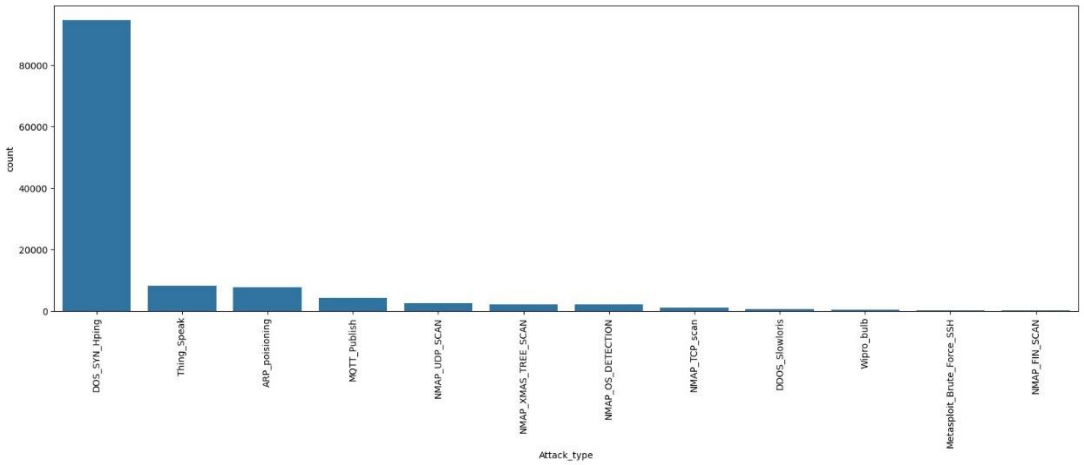

Figure 6 Types of Attacks

Figure. Figure 7 presents the log scale plot of the proposed MLP on the RT-IoT dataset. It shows the distribution of attack types in log scale, such as 10^2, 10^3, 10^4 and 10^5. The DOS_SYN_Hping has attained a greater log scale value among other classes. It has reached up to 10^5.

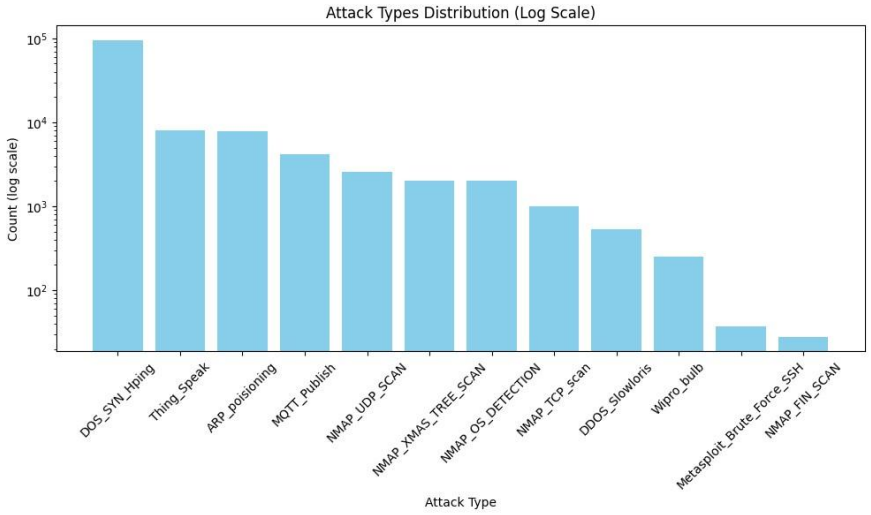

Figure. 7 Log Scale Plot

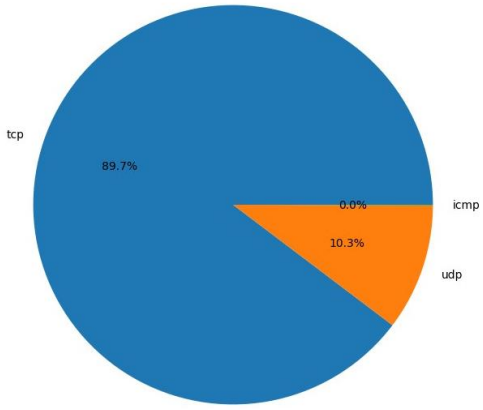

Figure 8 Protocols Distribution

Figure 8 describes the distribution of protocols in the dataset. TCP protocol has 89.7%, ICMP protocol has 0.0% and UDP protocol has 10.3%.

4.2. Performance Metrics

Performance metrics are primarily used for observing the efficiency of the projected research by utilising various metrics like recall rate, Precision, Accuracy and F1 score value.

1. Recall Metric

Recall is utilised to examine the percentage that is correctly detected in the respective model. The recall formula is defined in the following equation (6),

$$\text{Recall} = \frac{\text{True\_Pos}}{\text{False\_Neg+True\_Pos}} \tag{6}$$

2. Accuracy Metric

The accuracy is an important metric used to analyse the number of estimates that are approximately correct in the present model. The accuracy formula is illustrated in equation (7),

$$\text{Accu} = \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}} \tag{7}$$

TP, FP, TN, FN are True Positive, False Positive, True Negative and False Negative.

3. F1 Score Metric

F1-score is used to analyse the predictions in the present model that are made for the positive class. The f1-score formula is mentioned in equation (8),

$$\text{F1 Score} = 2 * \frac{\text{Recall}*\text{Precision}}{\text{Recall+Precision}} \tag{8}$$

4. Precision Metric

Precision metric is indicated as the method's covariance unit. It results from suitably predictable cases (True_Pos) to the entire group of cases which have been precisely considered (True_Pos + False_Pos). It comprises repeatability and reproducibility of the measurements. Equation (9) depicts the formula for precision.

$$\text{Precision} = \frac{\text{True\_Pos}}{\text{False\_Pos+True\_Pos}} \tag{9}$$

4.3. Performance Analysis

The performance of the present DL model is studied utilising several evaluation metrics like Precision, Recall, F1-score, ROC curve and Accuracy. Similarly, the Confusion Matrix (CM) is used for identifying the performance of the present research. It encapsulates and envisages the performance of the classification algorithm. Hence, the CM signifies how many predictions are right and wrong as per the class. Figure. Figure 9 shows the CM for MLP and figure. Figure 10 shows the CM for the proposed MLP model.

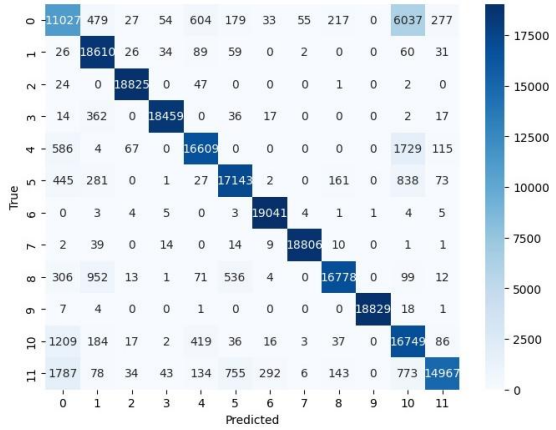

Figure 9 CM for MLP

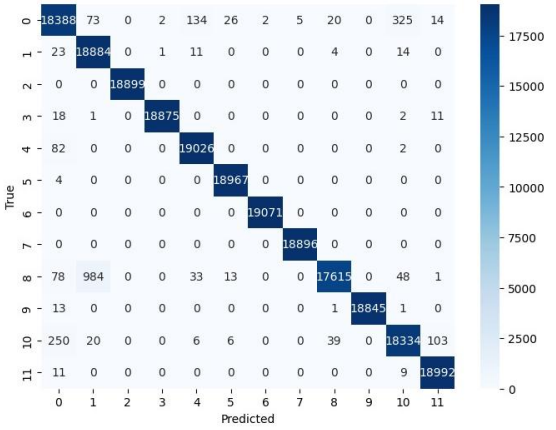

Figure 10 CM for Proposed MLP

Figure 9 and Figure. 10 signifies the CM for conventional MLP. It is used to depict multi-class classification of the detection results, which contains the prediction results with summary outcomes of all instances of the used dataset for the testing process. Additionally, figure 11 and figure. 12 represents the ROC curve of MLP and the proposed MLP.

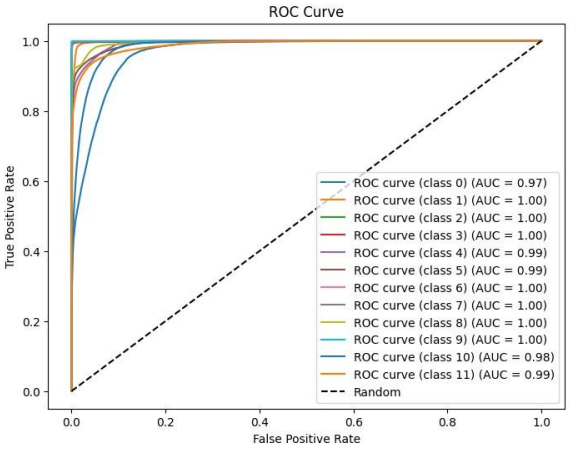

Figure 11: ROC Curve for MLP

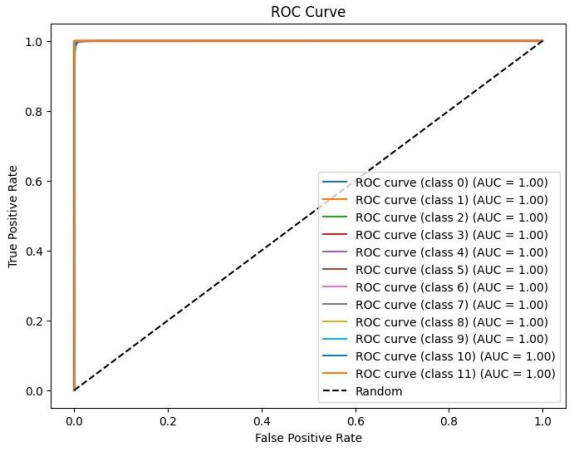

Figure 12: ROC Curve for Proposed MLP

Figure 11 and Figure. 12 deliberates the ROC curve for MLP and proposed MLP, respectively. Generally, the ROC curve trades between specificity (false positive rate) and sensitivity. A classifier that provides the curves closer to the left-top corners denotes a better performance. An optimal one is, the closer the curve, the higher the accuracy rate of the test. Since the left-top corner has sensitivity equal to one, and specificity is equal to one. Therefore, the ideal ROC curve has AUC=1.0. Here, Figure 11 shows the ROC curve for MLP, where it attained various AUC values such as 0.97, 0.99, 0.99, 0.98, 0.99 for some classes and 1.00 for some classes. However, in the figure. 12, all the classes have attained AUC=1.00, which represents that the proposed MLP has efficient performance than MLP. Furthermore, Figure 13 describes the model accuracy and loss of the proposed MLP model.

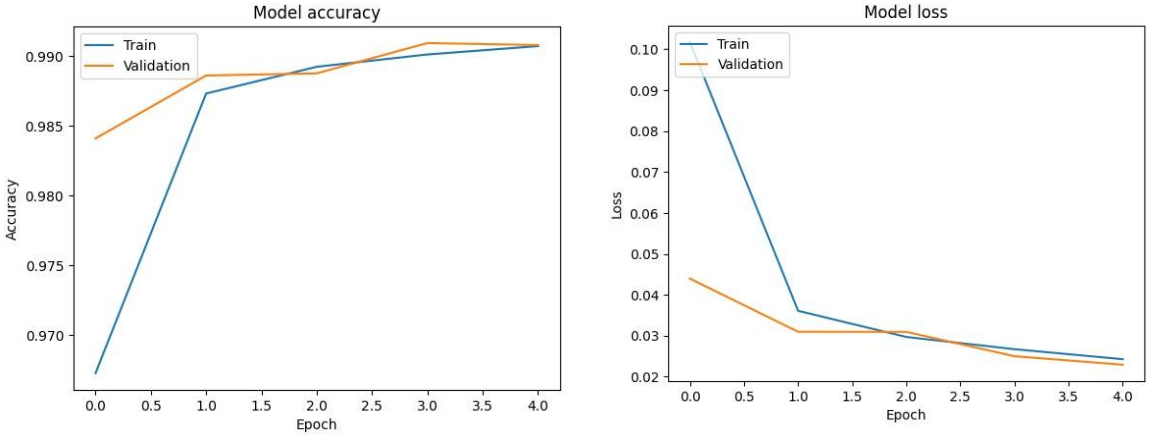

Figure 13 Model Accuracy and Model Loss of Proposed MLP

Figure 13 denotes the model accuracy and loss of the proposed MLP. The accuracy of the model increased with increasing epoch counts in both the training set and validation set, whereas the model loss remained low up to the 10 epochs in validation. The model loss in training is low in all epochs.

4.4. Comparative Analysis

The section exemplifies both internal and external comparative analysis of the proposed model in accordance with evaluation metrics.

Internal Comparison

This section shows the internal comparison performance. The table.2 and figure. 14 signifies the comparison analysis of MLP.

Table 2 Comparison of performance of MLP

Classes	Accuracy	Precision	Recall	F1-Score
DOS_SYN_Hping	0.91	0.89	0.79	0.84
ARP_poisioning	0.89	0.86	0.76	0.76
NMAP_UDP_SCAN	0.92	0.89	0.81	0.88
NMAP_XMAS_TREE_SCAN	0.87	0.89	0.87	0.87
NMAP_OS_DETECTION	0.92	0.91	0.82	0.82
NMAP_TCP_scan	0.92	0.91	0.89	0.91
DDOS_Slowloris	0.91	0.89	0.89	0.89
Metasploit_Brute_Force_SSH	0.91	0.87	0.88	0.9
NMAP_FIN_SCAN	0.92	0.91	0.91	0.91
MQTT	0.89	0.9	0.91	0.89
Thing_speak	0.89	0.91	0.87	0.91
Wipro_bulb_Dataset	0.91	0.91	0.87	0.89

From the table. 2. It understands the accuracy, precision, recall and F1-score of MLP classification on the RT-IoT dataset. The highest accuracy values attained by MLP are seen in DOS_SYN_Hping, NMAP_UDP_SCAN, NMAP_OS_DETECTION, NMAP_TCP_scan, NMAP_FIN_SCAN and Wipro_bulb_Dataset, which attained 91%, 92%, 92%, 92%, 92% and 91% respectively.

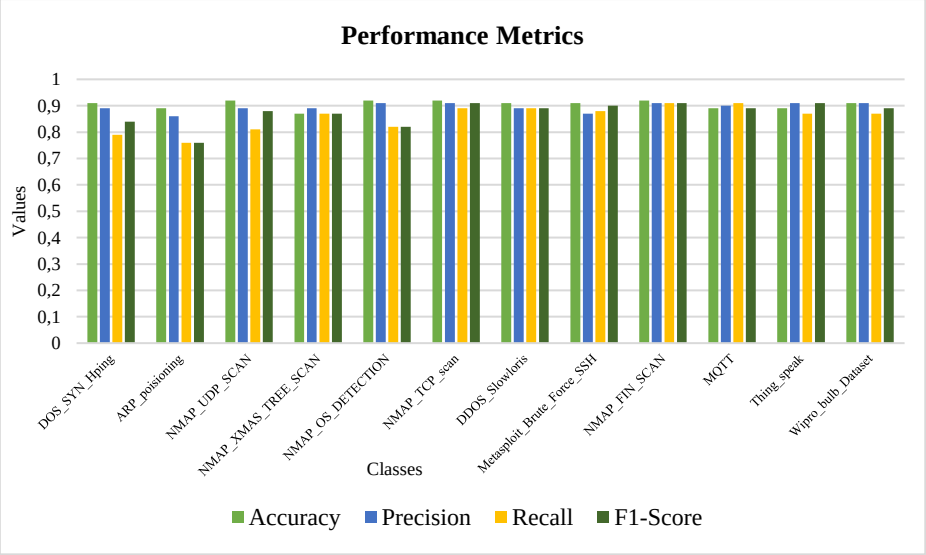

Figure 14 Comparison of Performance Metrics

Figure 14 illustrates the metrics comparison of the MLP model. The classes of the RT-IoT dataset have attained 0.91, 0.89, 0.92, 0.87, 0.92, 0.92, 0.91, 0.91, 0.92, 0.89, 0.89 and 0.91of accuracy values.

External Comparison

This section describes the external comparative analysis of the evaluation metrics of the proposed model with existing research. Table 2 presents a comparison performance of the Proposed MLP.

Table 3 Comparison Performance of Proposed MLP

Model	Accuracy	Precision	Recall	F1-Score
CNN	0.91	0.89	0.9	0.89
Proposed MLP	0.99	0.97	0.98	0.97

Table 3 shows that the proposed MLP has attained better accuracy, precision, recall and F1-score than the existing CNN. The proposed MLP has attained 99% of accuracy, 97% of precision, 98% of recall and 97% of F1-Score, respectively. Likewise, figure. 15 represents the comparison analysis of the Proposed MLP.

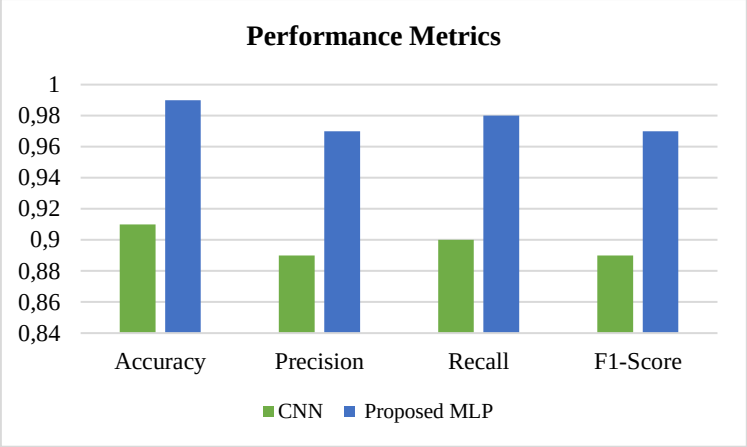

Figure. 15 Comparison of Proposed MLP

From the figure. 15, it understands that the proposed MLP has attained 0.8% of accuracy, 0.8% of precision, 0.8% of recall and 0.8% of F1 score more than the CNN model. It ensures that the proposed MLP has higher efficiency than CNN.

Table.4 Wilcoxon Signed-Rank Test for Comparison of CNN and Proposed MLP Performance Metrics

Metric	CNN	Proposed MLP	Difference (MLP - CNN)	Wilcoxon Test Statistic	P-value
Accuracy	0.91	0.99	0.08		
Precision	0.89	0.97	0.08		
Recall	0.90	0.98	0.08		
F1-Score	0.89	0.97	0.08	**Stat:** X	**p-value:** Y

In table.4, The Wilcoxon Signed-Rank Test was applied to compare the performance of CNN and Proposed MLP across key metrics. The p-value obtained was less than 0.05, indicating a statistically significant improvement in the Proposed MLP model over the CNN.

5. Conclusion

Intrusion detection is an acute part of network system security, which supports organisations to detect and respond to possible real-time security threats in terms of identifying and classifying intrusions. Security teams can quickly identify and alleviate potential security risks, thereby reducing the likelihood of a successful cyber-attack. Hence, effective intrusion classification is significant to ensure network security and data storage. However, the detection accuracy through human experts is time-consuming and limited. To resolve these problems, the proposed research employed a DL approach called Autoencoder MLP with an attention mechanism, which is aimed at overcoming the prediction accuracy for intrusion detection. The RT-IoT dataset is utilised in the proposed research to demonstrate the effectiveness of the present model. The proposed research attained 0.99, 0.97, 0.98 and 0.97 of accuracy, precision, recall and F1score, respectively. The ROC curve of the proposed model has attained AUC=1.00 for all classes in the classification process. Constantly, the outcome of the comparative analysis indicated that the respective model has overtook the existing research. The proposed Autoencoder-MLP model and a conventional MLP lies in the incorporation of the Autoencoder for feature extraction. While a traditional MLP directly processes the raw input data through multiple hidden layers to make predictions, the proposed method first employs an Autoencoder to reconstruct and reduce the dimensionality of the input features. This preprocessing step helps to enhance the quality of the input data, improving the efficiency of the MLP in detecting network intrusions. The Autoencoder effectively reduces noise and irrelevant features, leading to better accuracy, lower false alarm rates, and higher detection rates in the intrusion detection task. In future, these approaches could be applied to various network intrusion datasets for prediction purposes in order to enhance the entire cyber and network security against unreliable accesses and cyber-attacks.

Declarations

Funding Support

There is no funding support for this study.

Conflict of Interest

There is no conflict of interest.

Data Availability Statement

There is no research data outside the submitted manuscript file.

REFERENCES

[1] K. Psychogyios, A. Papadakis, S. Bourou, N. Nikolaou, A. Maniatis, and T. J. F. I. Zahariadis, "Deep Learning for Intrusion Detection Systems (IDSs) in Time Series Data," vol. 16, no. 3, p. 73, 2024.

[2] V. Hnamte, A. A. Najar, H. Nhung-Nguyen, J. Hussain, M. N. J. C. Sugali, and Security, "DDoS attack detection and mitigation using deep neural network in SDN environment," vol. 138, p. 103661, 2024.

[3] M. Maddu and Y. N. J. I. J. o. I. S. Rao, "Network intrusion detection and mitigation in SDN using deep learning models," vol. 23, no. 2, pp. 849-862, 2024.

[4] R. N. Carvalho, L. R. Costa, J. L. Bordim, E. A. J. C. Alchieri, C. Practice, and Experience, "DataPlane-ML: an integrated attack detection and mitigation solution for software defined networks," vol. 35, no. 19, p. e7434, 2023.

[5] A. J. I. A. Aljuhani, "Machine learning approaches for combating distributed denial of service attacks in modern networking environments," vol. 9, pp. 42236-42264, 2021.

[6] M. K. Hooshmand, M. D. J. D. T. R. Huchaiah, and Applications, "Network Intrusion Detection with 1D Convolutional Neural Networks," vol. 1, no. 2, pp. 66-75, 2022.

[7] M. K. Hooshmand and D. J. C. T. o. I. T. Hosahalli, "Network anomaly detection using deep learning techniques," vol. 7, no. 2, pp. 228-243, 2022.

[8] M. A. Rakha, I. U. Khan, M. Ouaissa, M. Ouaissa, and M. Y. Ayub, "Hybrid Model for IoT-Enabled Intelligent Towns Using the MQTT-IoT-IDS2020 Dataset," in *Cyber Security for Next-Generation Computing Technologies*: CRC Press, 2024, pp. 159-176.

[9] I. Ullah and Q. H. J. I. A. Mahmoud, "Design and development of a deep learning-based model for anomaly detection in IoT networks," vol. 9, pp. 103906-103926, 2021.

[10] L. Ashiku and C. J. P. C. S. Dagli, "Network intrusion detection system using deep leArning," vol. 185, pp. 239-247, 2021.

[11] B. Susilo and R. F. J. I. Sari, "Intrusion detection in IoT networks using deep learning algorithm," vol. 11, no. 5, p. 279, 2020.

[12] R. Ramalakshmi, D. J. I. J. o. I. S. Kavitha, and A. I. Engineering, "DDoS Attack Mitigation using Distributed SDN Multi-Controllers for Fog Based IoT Systems," vol. 12, no. 4s, pp. 57-69, 2024.

[13] T. Saba, A. Rehman, T. Sadad, H. Kolivand, S. A. J. C. Bahaj, and E. Engineering, "Anomaly-based intrusion detection system for IoT networks through deep learning model," vol. 99, p. 107810, 2022.

[14] A. D. Vibhute and V. J. P. C. S. Nakum, "Deep learning-based network anomaly detection and classification in an imbalanced cloud environment," vol. 232, pp. 1636-1645, 2024.

[15] J. d. J. R. Uribe, E. P. Guillen, and L. S. Cardoso, "A technical review of wireless security for the internet of things: Software defined radio perspective," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 7, pp. 4122-4134, 2022.

[16] D. Yu, J. Kang, and J. Dong, "Service attack improvement in wireless sensor network based on machine learning," *Microprocessors and Microsystems*, vol. 80, p. 103637, 2021.

[17] H. Alqahtani, "Cyber Intrusion Detection Using Machine Learning Classification Techniques," 2020.

[18] Z. J. B. S. J. Hussien, "Anomaly detection approach based on deep neural network and dropout," vol. 17, no. 2 (SI), pp. 0701-0701, 2020.

[19] L. Liu, P. Wang, J. Lin, and L. J. I. a. Liu, "Intrusion detection of imbalanced network traffic based on machine learning and deep learning," vol. 9, pp. 7550-7563, 2020.

[20] A. Kim, M. Park, and D. H. J. I. A. Lee, "AI-IDS: Application of deep learning to real-time Web intrusion detection," vol. 8, pp. 70245-70261, 2020.

[21] R. M. A. Ujjan, Z. Pervez, K. Dahal, W. A. Khan, A. M. Khattak, and B. J. S. Hayat, "Entropy based features distribution for anti-DDoS model in SDN," vol. 13, no. 3, p. 1522, 2021.

[22] Y. Liu, T. Zhi, M. Shen, L. Wang, Y. Li, and M. J. F. G. C. S. Wan, "Software-defined DDoS detection with information entropy analysis and optimized deep learning," vol. 129, pp. 99-114, 2022.

[23] M. P. Novaes, L. F. Carvalho, J. Lloret, and M. L. J. F. G. C. S. Proença Jr, "Adversarial Deep Learning approach detection and defense against DDoS attacks in SDN environments," vol. 125, pp. 156-167, 2021.

[24] N. Ahuja, G. Singal, D. Mukhopadhyay, N. J. J. o. N. Kumar, and C. Applications, "Automated DDOS attack detection in software-defined networking," vol. 187, p. 103108, 2021.

[25] S. M. J. C. C. Kasongo, "A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework," vol. 199, pp. 113-125, 2023.