Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7766317

# NETWORK INTRUSION DETECTION SYSTEM USING DEEP LEARNING MODELS TO CAPTURE CYBER ATTACKS

**Mrs. Priya Vengatesh**
Ph.D. Scholar,

**Dr. R.Kannan**
MCA., M.Phil., Ph.D., Associate Professor, Department of Computer Science
Sri Ramakrishna Mission Vidyalaya College of Arts and Science, Coimbatore – 641 020.

**Abstract**

Cyber threats are increasing and to mitigate cyber attacks and threats, intrusion detection system is introduced. Intrusion detection systems are widely used to capture the deviating patterns in the network traffic. Due to dynamic nature of changing patterns of threats and attacks, an efficient model is required to update the attacks and patterns present in the network traffic data. Many machine learning models are deployed to learn the traffic patterns but traditional models largely suffer from high traffic volume and high dimensional features. This paper proposes a deep learning model which is resilient to capture network intrusions with better learning ability. The effectiveness of the proposed deep learning model is demonstrated using CICIDS2017 dataset and the performance of the proposed model achieved accuracy of 99.7% over other machine learning models.

## 1. Introduction

With the development of networking, hardware and software capabilities, cyber attacks are increasing. Presently cyber attacks are evolving which are complicated and challenging to detect. Since attackers use sophisticated methods to evade detection and a novel intrusion detection system is required to prevent and capture network intrusions. Many machine learning models such as Support Vector Machine, Random Forest, decision tree, Neural Network (Zou et al., 2009) etc are currently been used to detect intrusions in the network (Ahmad et al., 2019, Prachi & Sharma, 2019; Asad et al., 2020). An intrusion detection system can be categorized into host based and network based intrusion detection. The host based and network based IDS are further grouped into anomaly based, signature based and hybrid (Aydin et al., 2009).

The large volume of network traffic data requires highly sophisticated learning models to process high volume data and to handle high dimensional feature sets. Large volume of data and features increase the computational complexity and learning time in order to classify traffic into normal and attack. Further, the presence of different types of attacks and intrusions in the network traffic demands an efficient model to capture nuances of intrusions. Deep learning models are currently used for its scalability, high learning capability, and feature engineering ability in image processing, language processing, recommendation systems and fault detection (Chhajer et al, 2022). Contrasting shallow learners, deep learning models are more robust in extracting the feature information and learning of features. The dimensions of large data often affect the learning capabilities leading to poor detection of attacks and exploitation from a

dataset having large number of features. Deep learning models overcome the limitations of machine learning models through extracting features that map the intrusion type and attacks. Machine learning models require domain experts to bring down the feature dimension complexity through feature selection. Deep learning models are capable of training non-linear data which promise accurate prediction of network intrusion and its sub categories through generalization of new attack variants (Mighan & Kahani, 2021).

Deep learning models have different networks such as Convolution Neural Networks (CNN), Long Short Term Memory Networks (LSTM), Recurrent Neural Networks (RNN), Generative Adversarial Networks (GAN), Multilayer Perceptrons (MLP), Self Organizing Maps (SOM) and Deep Belief Networks (DBN) which are capable of different properties to utilize the features. DL models have layers and each layer receives an input from its preceding layer. DL models process the raw features and propagate the important feature information to the next layers, which are finally used to classify or predict the labels. The main advantage of deep learning models is the scalability, unsupervised self-learning to improve accuracy, supports parallel and distributed training and automatic feature generation. The one disadvantage is that it requires high knowledge on network topologies and training parameters.

## 2. Related works

Deep learning models have attracted many scholars and researchers towards developing Intrusion Detection Systems using vast amount of data. Since traditional machine learning models do not handle large data, deep learning models are primarily chosen for its stability and scalability. (Qazi et al., 2022) proposed a one-dimensional CNN to classify network intrusions. The proposed model has five convolution layers and five dense layers and softmax layer to predict the probabilities of intrusion types. The proposed model achieved 98.96% accuracy for multiclass problem on CICIDS2017 dataset. A hybrid deep learning model was proposed by (Aldallal, A. 2022) to improve the detection rate of IDS based on GRU and LSTM. The relevant features are selected using Pearson Correlation method. The proposed Cu-LSTMGRU is combines LSTM and GRU with a computing unit called MPDM. The information that are important to classify are passed on the next layer through the update gate which information that are no longer required are cast-out through reset gate. The proposed model achieved highest accuracy of 97.76% on CICIDS2018 reducing the false alarm rate. The efficiency of IDS can be enhanced if the model can handle large volume of data, offer feature selection and have good performance over detection of attacks. To meet such requirements, (Adefemi Alimi et al., 2022) proposed a refined LSTM for denial-of-service attack detection. RLSTM has the advantage of preventing back propagation errors and address the problem of vanishing gradients. The proposed RLSTM showed accuracy of 99.2% for detecting dos-attack in CICIDS2017 dataset and 98.6% in NSL-KDD dataset. (Azzaoui et al., 2022) demonstrated the efficiency of deep neural networks using four layers architecture to detect intrusions in the network. The model show better performance with low false alarm rate and high accuracy of 99.43% on CICIDS2017 and 99.63% on NSL-KDD dataset.

(Jamil & Kim., 2021) proposed an ensemble based learning and prediction of anomaly in network traffic using automated machine learning. The proposed model use Bayesian optimization of parameter tuning and Kalman filter model for prediction. The proposed method is compared with one to five different layers of DNN and the proposed method show high

performance with accuracy of 97.02% on CICIDS2017 and 98.8% in UNSWNB15 dataset. (Sahu et al., 2021) proposed a hybrid deep learning model by combining CNN and LSTM to detect intrusion in the network. The purpose of the hybrid model is to use the automated feature learning from CNN and to classify intrusions using LSTM. The proposed model has four convolution layers with LeakyReLU as activation function. The selected features are fed into the LSTM layers for classification of intrusions. The model achieved a detection accuracy of 96% for multi attack types. Similarly a hybrid model using CNN and LSTM was proposed by (Sun et al., 2020) to detect intrusion in the network. The hybrid model captures the temporal and spatial features of the network traffic. The network architecture contains two convolution layers and two LSTM layers, which classified the intrusions correctly. The model achieved 98.67% of overall accuracy. The advantage of using LSTM layers is that settles the gradient exploding and disappearance while training in sequence. Also, (Kim et al., 2020) proposed IDS based on CNN-LSTM and the model performance is evaluated using three datasets. The network consists of two CNN layers and three LSTM layers, the data from output of CNN layers are fed into LSTM. The first layer of LSTM is forward direction and the second layer of LSTM is bidirectional and the last layer is DNN which accumulates the forward and backward cells of second layer LSTM. The proposed model show low performance of 83% on CICIDS2017 dataset and 91% in CSIC-2010 dataset. The low performance of the model is due to overfitting as the model has 14000 trainable parameters. (Kaur & Singh, 2020) proposed a hybrid deep learning model using RNN for network anomaly classification. The proposed model not only classifies network attacks but also generates signatures to infer by the IDS. Combining the signatures and the classifier, the proposed D-sign IDS outperforms other models with accuracy of 99.1% on CICIDS2017 and 99.14% on NSL-KDD datasets. (Sethi et al., 2021) proposed a multi-agent deep reinforcement learning based IDS for attack classification using attention mechanism. The network of ten layers and five deep sequential layers are studied and the performance of the model on CICIDS2017 is 98.7% and 97.4% on NSL-KDD dataset.

## 3. Methodology

### 3.1 CNN Architecture

Convolution Neural Networks (CNN) fall under deep learning (DL) architecture and similar to Neural Networks with stacked layers. The term convolution involves a convolution block in the neural network. CNN represents two dimensional inputs to categorical output for classification task and to real integers for regression task. A Neural Network typically requires handcrafted features to produce accurate results while CNN uses raw data and process the input across many convolution layers. The input to CNN is a vector that represents the network traffic X. The CNN is designed to learn a set of parameters Θ that map the input to the prediction C (attack classes) and it can be represented as,

$$C= F(X \mid \Theta) = f_h(\ldots f_2(f_1(X|\Theta_1) \mid \Theta_2)| \Theta_h) \quad (1)$$

where h is the number of hidden layers and for the $i^{th}$ layer in the convolution layer can be represented as,

$$C_1=f_i(X_i|\Theta_i) = A(W*X_i + b), \Theta_i = [W, b] \quad (2)$$

Where * is the dot product (convolution function) with the input features, $X_i$ is the two dimensional input matrix of N features, W is the one dimensional kernels for extracting new

features from input vector and b is the bias vector and A is the activation function. Many Pooling layer is used between convolution layers that summarize the features by calculating maximum or average of the input. The output of the convolution layer is flattened and passed on to fully connected layers and it can be represented by equation 3 and for multiclass classification, softmax activation function is used on the output layer, where the each neuron points the class membership of the input samples.

$$C_l = f_i(X_i|\Theta_i) = A(W X_i + b),\ \Theta_i = [W, b] \quad (3)$$

CNN is made up of three main layers, convolution layer, pooling layer and fully connected layer. The features are extracted from convolution layer and pooling layer while fully connected layer relates the features extracted to the output attack classes. Convolution refers to a linear function used to extract features. The non linear activation functions used in the convolution layer is ReLU, sigmoid and tanh function. The pooling layer is used to reduce the feature dimensions by aggregating feature information from different kernels. Max pooling and average pooling are the two types of pooling available where max pooling extract features with respect to kernel size while discarding others. The fully connected layer is last layer where the features are transformed into 1D vector which is fully connected to each output class through weights. The last layer activation function vary according to the type of problem i.e., binary, multiclass or regression. For binary classification & multiclass classification sigmoid activation and for regression linear or identity activation function is used. The parameters are adjusted with respect to the classification error rates which are back propagated while training through minimizing the loss function.

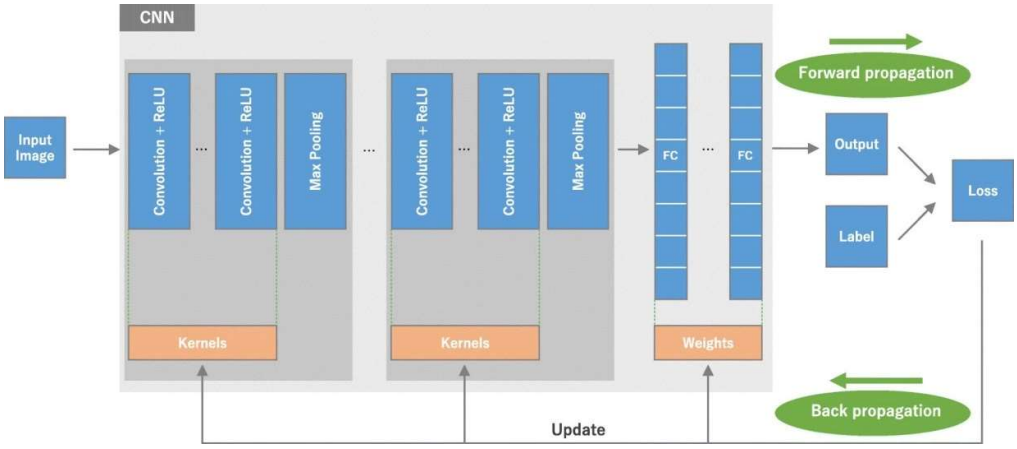


Figure 1 CNN Architecture

CNN architecture is characterized by set of hyperparameters which refers to network structure and training hyperparameters. The hyperparameters of network structure refers to layer number, units in each layer, kernel size, strides, pooling and activation functions while the training hyperparameters refers to learning rate, batch size, momentum, epochs, optimizer and patience for early stopping. The parameters and hyperparameters of typical CNN are given in Table 1.

Table 1 Parameters and Hyperparameters of CNN

| | Parameters | Hyperparameters |
|---|---|---|

| | | |
|---|---|---|
| Convolution layer | Kernels | Kernel size, number of kernels, stride, activation function |
| Pooling layer | - | Filter size, method, stride |
| Fully Connected layer | Weights | Number of weights, Activation function |
| Others | - | Learning rate, optimizer, loss function, batch size, epochs, regularization |

### 3.2 Proposed 1D CNN

The performance of the CNN depends on the architecture of the network design which differs according to different problem and sizes. It is difficult to find the suitable hyperparameters for a particular problem and adjusting the hyperparameters value directly affects the model performance. Optimizing network parameters and finding the right combination of hyperparameters improve the model performance otherwise the model might not classify intrusions effectively. Poor model performance indicates that the model stability, processing time and computing resources is affected. To effectively classify intrusions, 1D CNN is constructed separately for binary and multiclass problem. 1D CNN refers to the kernel that slides in one dimension. Three convolution layers are added after the input layer followed by two fully connected layers and an output layer. The first layer has 128 neurons and second convolution layer has 64 neurons and the second layer is interlaced with max pooling layer with pool size of 3 and kernel size of 3 x 3 with activation function relu. The third convolution layer has 32 neurons with kernel size of 3 x 3. To avoid overfitting a dropout layer is added to the third convolution layer in order to preserve the feature information. The output of the last convolution layer is flattened and passed on to fully connected layer. The fully connected layer has 10 neurons with softmax activation function predicts the network intrusion types. The loss function is given in equation 4.

$$\text{Loss} = -\sum_{i=1}^{k} y_j \log (y`_i) \quad (4)$$

where k is the number of classes, y is the actual class and y` is the predicted class.

Table 2 Proposed 1D CNN for binary classification

| Model | Layer | Output shape | Parameters | Filters | Kernel Size | Activation |
|---|---|---|---|---|---|---|
| ID CNN | Conv1d | (67,128) | 384 | 128 | 3 | relu |
| | Conv1d | (66,64) | 16448 | 64 | 3 | relu |
| | Dropout | (66,64) | 0 | | | |
| | MaxPooling1D | (33,64) | 0 | | | |
| | Conv1d | (32,64) | 8256 | 64 | 3 | relu |

| | | | | | | |
|---|---|---|---|---|---|---|
| | Dropout | (32,64) | 0 | | | |
| | Flatten | (2048) | 0 | | | |
| | Dense | (10) | 20490 | | 3 | relu |
| | Dense | (2) | 22 | | | softmax |
| Total Params: 45600<br>Trainable Params: 45600<br>Non trainable params: 0 | | | | | | |

Table 3 Proposed 1D CNN for multiclass classification

| **Model** | **Layer** | **Output shape** | **Parameters** | **Filters** | **Kernel Size** | **Activation** |
|---|---|---|---|---|---|---|
| ID CNN | Conv1d | (67,128) | 384 | 128 | 3 | relu |
| | Conv1d | (66,64) | 16448 | 64 | 3 | relu |
| | Dropout | (66,64) | 0 | | | |
| | MaxPooling1D | (33,64) | 0 | | | |
| | Conv1d | (32,64) | 8256 | 64 | 3 | relu |
| | Dropout | (32,64) | 0 | | | |
| | Flatten | (2048) | 0 | | | |
| | Dense | (10) | 20490 | | 3 | relu |
| | Dense | (4) | 44 | | | softmax |
| Total Params: 45622<br>Trainable Params: 45622<br>Non trainable params: 0 | | | | | | |

## 4. Experiment and analysis

### 4.1 Dataset

The proposed 1D convolution network performance is evaluated on CICIDS2017 dataset. The CICIDS2017 dataset contains real-time network traffic data which captured network traffic over five day period. This dataset contains latest attacks that resemble the real world data and also contain labeled flows for network traffic analysis based on time stamp, source, and destination IPs, source and destination ports, protocols and attack (Iman

Sharafaldin et al., 2018). The normal traffic is labeled as benign and 15 types of attacks types are labeled. The attacks includes Brute Force, Heart Bleed, Botnet, DoS, DDoS,Web, and Infiltration Attack. The entire dataset comprising five day traffic data has a total of 2,299,308 instances which require high computational capability. To test the proposed model, Thursday morning working hour's dataset is utilized for the study. The Thursday dataset has 170368 instances, 78 features and 1 label column. Around 10 features that have zero values are removed and rest of the features is included in the study. The final dataset set contains 68 features and 1 class label with three attack types and benign traffic.

Table 4 CICIDS2017 dataset

| Day | Type | Size |
|---|---|---|
| Monday | Normal | 11GB |
| Tuesday | Tuesday | 11GB |
| Wednesday | Normal + Dos + Heartbleed Attacks | 13GB |
| Thursday | Normal + XSS + Web Attack + Infiltration | 7.8GB |
| Friday | Normal + Botnet + PortScan + DDoS | 8.3GB |

### 4.2 Preprocessing

The categorical features present in the CICIDS2017 dataset is converted to nominal values and the numerical features are standardized using Z-score normalization. Since each feature have different ranges which affects the training of the module. The normalization helps to keep the range between 0 and 1 which improves the training in a better way. The normalization formula is given in equation 5. Some features have infinite numbers and such features are selected and the infinite values are replaced with zeros.

$$x_i = \frac{x_i - \mu}{\sigma} \quad (5)$$

where, μ is mean and σ is the standard deviation.

### 4.3 Evaluation Metrics

The performance of classification model is summarized and visualized using confusion matrix. The confusion matrix for binary classification is given in Table 5. It represents the actual values and predicted values. The performance is interpreted using TP, TN, FP and FN where TP refers to True Positives which represent the number of positive samples correctly classified as Positives, TN refers to True Negatives which represent the number of negative samples classified correctly as Negatives, FP refers to False Positives which represent the number of Negative samples incorrectly classified as Positive and FN refers to False Negatives which represent the number of Positive samples incorrectly classified as Negative. Accuracy, Sensitivity, specificity, Precision and F-Score are the metrics that explains the classification model's performance. The dataset is partitioned into testing and training set in the ratio 70:30 for binary classification and 80:20 ratio for multiclass classification.

Table 5 Confusion Matrix

| Actual \ Predicted | Positive | Negative |
|---|---|---|
| Positive | True Positive<br>TP | False Positive<br>FP |
| Negative | False Negative<br>FN | True Negative<br>TN |

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN}$$

$$\text{Sensitivity} = \frac{TP}{TP+FN}$$

$$\text{Specificity} = \frac{TN}{TN+FP}$$

$$\text{F1-score} = \frac{2TP}{2*TP + (FP+FN)}$$

## 5. Results and Discussion

### 5.1 Evaluation of the proposed model -Binary classification

Using CICIDS2017 dataset, the performance of the proposed model is evaluated. The metrics used to measure the performance are discussed in section 4.3. Developing an intrusion detection model using deep learning has been the hot trend in network security. Motivated by the computational advantage of 1D CNN over 2D CNN and feature learning capability, this study proposed a 1D CNN for capturing network intrusions or attacks. The proposed model is validated for binary class problem and multi-class problem. Binary classification involves two classes namely benign and intrusion while multi-class classification have four classes namely benign, web attack-Brute force, web attack-XSS and web attack-Sql injection.

**Table 6 Performance of the model for Binary classification**

| Class | Accuracy | Precision | Recall | F1-Score | FPR |
|---|---|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| 0 | 0.997 | 0.999 | 0.997 | 0.998 | 0.018 |
| 1 | 0.997 | 0.820 | 0.981 | 0.894 | 0.002 |
| Average | **99.7** | **90.9** | **98.1** | **94.6** | **0.01** |

The confusion matrix (Figure 2) shows true positives of 33541 instances, true negative of 431 instances, 94 instances of false positive and 8 instances of false negative. The performance of the model achieved an accuracy of 99.7%, precision of 90.9%, recall of 98.1%, f1 score of 94.6% and false positive rate of 0.01 Table 6. Accuracy of the model accounts for the total instances correctly classified, out of 34074 instances, the model correctly classified 33972 instances. Precision refers to the ratio of positive cases correctly classified as positive, in intrusion detection benign traffic is correctly classified as benign to about 90.9% while recall refers to ratio of the positive samples correctly classified as positive to the total number of positive cases, the 98.1% of recall shows that the proposed model correctly classified 98.1% of benign cases (33541) to the total number of benign cases (33635). F1 score refers to the harmonic mean of precision and recall, the F1 score of 99.8% refers to the good balance between precision and recall. The model achieved FPR of 0.01 for binary classification which refers to that the attack classes are correctly classified into negative class which lowers the incorrect classification of attack class as benign. Figure 3 represents the validation loss and Figure 4 represents the accuracy of the proposed model for binary classification.

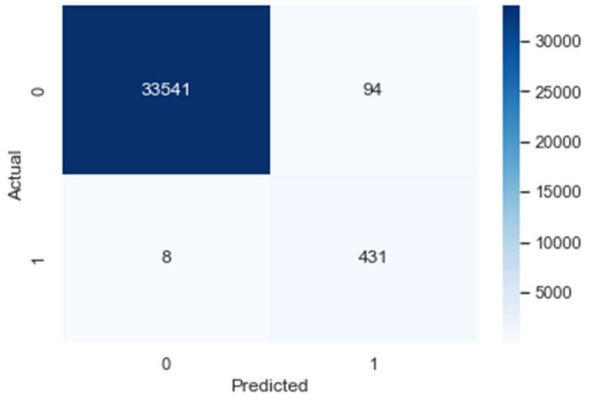


Figure 2 confusion matrix for binary classification

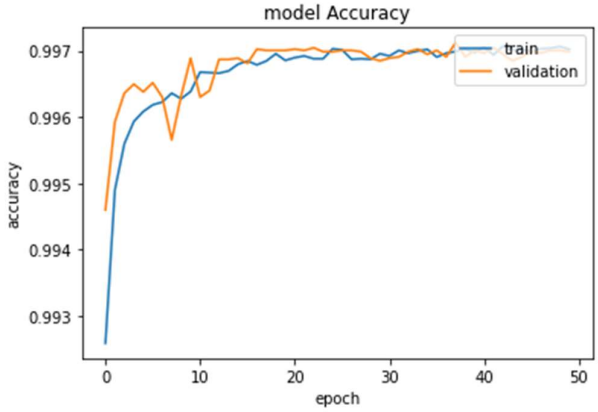


Figure 3 Accuracy of the proposed model for binary classification

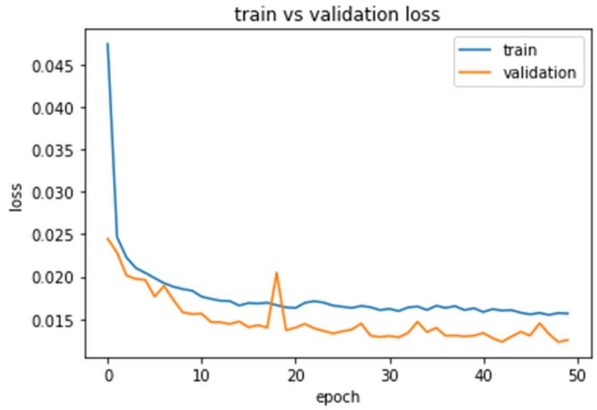


Figure 4 Validation loss of the proposed model for binary classification

**5.2 Evaluation of the proposed model - Multiclass classification**

The CICIDS2017 dataset has benign, web attack-Brute force, web attack-XSS and web attack-Sql injection classes. The performance of the proposed model is evaluated on different attack classes. The performance of the model achieved an accuracy of 99.6% for brute force, 99.9% for XSS, 99.9% for Sql injection and for benign 99.6%. The average model performance for all the classes show accuracy of 99.7%, precision of 93%, recall of 86.7%, f1 score of 87.0% and false positive rate of 0.001 (Table 7). Accuracy of the model accounts for the total instances correctly classified, out of 51070 instances, the model correctly classified 50909 instances and about 161 instances are incorrectly classified. Precision refers to the ratio of positive cases correctly classified as positive, in intrusion detection benign traffic is correctly classified as benign to about 99.9%, for brute force, XSS and sql injection the precision is 72.1%, 100% and 100%. Recall refers to ratio of the positive samples correctly classified as positive to the total number of positive cases, the recall for brute force, XSS and sql injection is 99.5%, 50% and 97.8%. The lower recall of 50% for XSS is due lower number of instances. F1 score refers to the harmonic mean of precision and recall, the F1 score of 99.8% and 98.3% for benign and sql injection refers to the good balance between precision and recall while attack class brute force have f1 score of 83.6% and XSS class have 66.6% of f1 score. The model achieved FPR of 0.003 for benign class and brute force which refers to that the normal and attack class brute force are correctly classified into positive and negative class. The null value of FPR for XSS and sql injection is attributed to the lower number of prediction samples.

**Table 7 Performance of the model for Multiclass classification**

| Class | Accuracy | Precision | Recall | F1-Score | FPR |
|---|---|---|---|---|---|
| Benign | 0.996 | 0.999 | 0.996 | 0.998 | 0.003 |
| Brute force | 0.996 | 0.721 | 0.995 | 0.836 | 0.003 |
| XSS | 0.999 | 1 | 0.5 | 0.666 | 0 |
| Sql injection | 0.999 | 1 | 0.978 | 0.983 | 0 |
| Average | **99.7** | **93** | **86.7** | **87.0** | **0.001** |

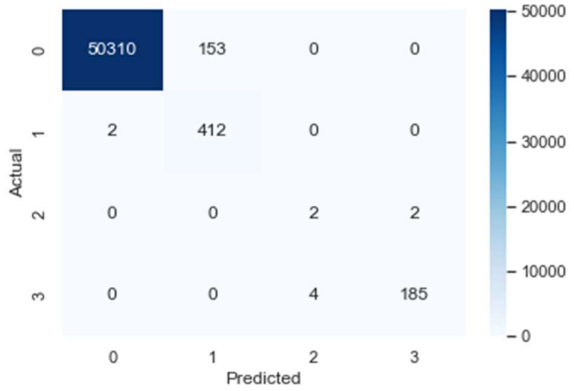


Figure 5 confusion matrix for multiclass classification

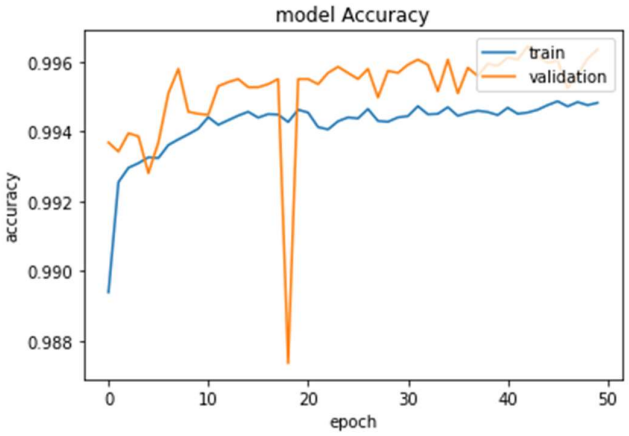


Figure 6 Accuracy of the proposed model for binary classification

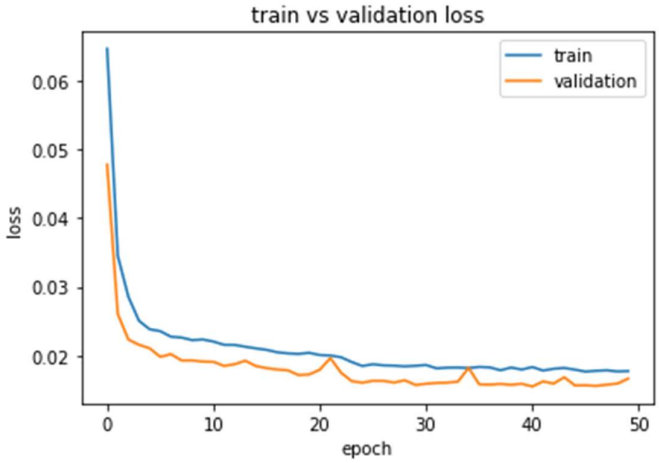


Figure 7 Validation loss of the proposed model for multiclass classification

To demonstrate the efficiency of the proposed model, the performance of model is compared with state-of-art deep learning methods. Many deep learning models are currently studied to classify network intrusions and models are trained for both binary and multiclass problems. The proposed model consists of three convolution layers and two dense layers with max pooling and single dropout (0.3) in the third convolution layer. In order to preserve the feature information the dropout layer is added to the third layer. The model achieved highest accuracy of 99.7% over 1DCNN proposed by (Qazi et al., 2022). The proposed model equally performed well with an accuracy of 99.7% with (Aldallal, A., 2022) who proposed Cu-LSTMGRU, a gated recurrent network which require high computing power. Compared to CNN, CNN-LSTM models have more parameters to handle and are sensitive to initial weights which could affect the training on spatial features. The proposed model performed good learning on spatial features as the number of parameters is less compared to CNN-LSTM, which facilitate more learning on feature maps which outperformed (Sahu et al., 2021; Sun., et al., 2020; Kim et al., 2020). RLSTM model which falls under RNN, requires more training data and due to its nature of back propagation, this model might suffer from vanishing or exploding problem and also suffers on training long sequences. The proposed model produced higher accuracy of 99.7% than RLSTM (99.22%) and RNN network (99.1%). The proposed CNN model outperforms RNN models of (Adefemi Alimi et al., 2022) and (Kaur & Singh, 2020). The validation loss and accuracy of the model is given in Figure 6 and Figure 7 which demonstrate that 1DCNN is better in terms of accuracy, recall, precision and f1 score than other models discussed in the literature.

Table 8 comparison of proposed model with deep learning methods

| Author | Year | Method | Accuracy% |
|---|---|---|---|
| Qazi et al., | 2022 | 1DCNN | 98.96 |
| Aldallal, A. | 2022 | Cu-LSTMGRU | 99.70 |
| Adefemi Alimi et al., | 2022 | RLSTM | 99.22 |
| Azzaoui et al., | 2021 | DNN | 99.43 |
| Sethi et al., | 2021 | MARL | 98.70 |
| Jamil & Kim., | 2021 | Ensemble Model | 97.02 |
| Sahu et al., | 2021 | CNN-LSTM | 96.0 |

| | | | |
|---|---|---|---|
| Sun et al., | 2020 | CNN-LSTM Hybrid | 98.67 |
| Kim et al., | 2020 | CNN-LSTM | 93.0 |
| Kaur & Singh., | 2020 | RNN | 99.1 |
| | | 1DCNN-binary | **99.7** |
| **Proposed** | 2022 | 1DCNN-Multiclass | **99.7** |

## Conclusion

This paper proposed a CNN model with one dimension for network intrusion detection. The proposed model is evaluated using CICIDS2017 dataset. The pre-processed data set was trained on 1DCNN model constructed with three convolution layers and two dense layers, max pooling and fully connected layer. The performance of the CNN model is interpreted with performance metrics such as accuracy, precision, recall and f1 score. The 1DCNN model achieved 97% of accuracy, 90.9% of precision, 98.1% of recall, 94.6 of f1 score and 0.01 of FPR for binary. In multiclass classification, the model accomplished average accuracy of 99.7%, 93% of precision, 86.7% of recall and 87.0% of f1 score with 0.001 of FPR. Based on the results, the proposed 1D CNN model is efficient for network intrusion detection and therefore it can be employed for network attack detection. As a future work, the effect of feature selection on network attack classification will be explored with other state of CNN by fine tuning the network parameters to achieve optimum performance in terms of large network traffic volume and large number of network traffic features. Also, the attack types are evolving and it is necessary to include newer threats and attack types to effectively detect intrusions.

## References

1. Adefemi Alimi, K. O., Ouahada, K., Abu-Mahfouz, A. M., Rimer, S., & Alimi, O. A. (2022). Refined LSTM Based Intrusion Detection for Denial-of-Service Attack in Internet of Things. Journal of Sensor and Actuator Networks, 11(3), 32.
2. Ahmad, I., Basheri, M., Iqbal, M. J., & Rahim, A. (2018). Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. IEEE access, 6, 33789-33795.
3. Aldallal, A. (2022). Toward Efficient Intrusion Detection System Using Hybrid Deep Learning Approach. Symmetry, 14(9), 1916.
4. Asad, M., Asim, M., Javed, T., Beg, M. O., Mujtaba, H., & Abbas, S. (2020). Deepdetect: detection of distributed denial of service attacks using deep learning. The Computer Journal, 63(7), 983-994.
5. Atefinia, R., & Ahmadi, M. (2021). Network intrusion detection using multi-architectural modular deep neural network. The Journal of Supercomputing, 77, 3571-3593.
6. Aydin, M. A., Zaim, A. H., & Ceylan, K. G. (2009). A hybrid intrusion detection system design for computer network security. Computers & Electrical Engineering, 35(3), 517-526.
7. Azzaoui, H., Boukhamla, A. Z. E., Arroyo, D., & Bensayah, A. (2022). Developing new deep-learning model to enhance network intrusion classification. Evolving Systems, 13(1), 17-25.

8. Chhajer, P., Shah, M., & Kshirsagar, A. (2022). The applications of artificial neural networks, support vector machines, and long–short term memory for stock market prediction. Decision Analytics Journal, 2, 100015.
9. Fernandez, G.C.; Xu, S. A Case Study on using Deep Learning for Network Intrusion Detection. In Proceedings of the MILCOM 2019—2019 IEEE Military Communications Conference (MILCOM), Norfolk, VA, USA, 12–14 November 2019; pp. 1–6.
10. Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization", 4th International Conference on Information Systems Security and Privacy (ICISSP), Portugal, January 2018
11. Jamil, F., & Kim, D. (2021). An ensemble of prediction and learning mechanism for improving accuracy of anomaly detection in network intrusion environments. Sustainability, 13(18), 10057.
12. Kaja, Nevrus, Adnan Shaout, and Di Ma. "An intelligent intrusion detection system." Applied Intelligence 49 (2019): 3235-3247.
13. Kaur, S., & Singh, M. (2020). Hybrid intrusion detection and signature generation using deep recurrent neural networks. Neural Computing and Applications, 32, 7859-7877.
14. Kim, A.; Park, M.; Lee, D.H. AI-IDS: Application of Deep Learning to Real-TimeWeb Intrusion Detection. IEEE Access 2020, 8,70245–70261.
15. Louati, F., & Ktata, F. B. (2020). A deep learning-based multi-agent system for intrusion detection. SN Applied Sciences, 2(4), 675.
16. Mhawi, D. N., Aldallal, A., & Hassan, S. (2022). Advanced feature-selection-based hybrid ensemble learning algorithms for network intrusion detection systems. Symmetry, 14(7), 1461.
17. Mighan, S. N., & Kahani, M. (2021). A novel scalable intrusion detection system based on deep learning. International Journal of Information Security, 20, 387-403.
18. Prachi, H. M., & Sharma, P. (2019). Intrusion detection using machine learning and feature selection. International Journal of Computer Network and Information security, 11(4), 43-52.
19. Qazi, E. U. H., Almorjan, A., & Zia, T. (2022). A One-Dimensional Convolutional Neural Network (1D-CNN) Based Deep Learning System for Network Intrusion Detection. Applied Sciences, 12(16), 7986.
20. Sahu, A. K., Sharma, S., Tanveer, M., & Raja, R. (2021). Internet of Things attack detection using hybrid Deep Learning Model. Computer Communications, 176, 146-154.
21. Sethi, K., Madhav, Y. V., Kumar, R., & Bera, P. (2021). Attention based multi-agent intrusion detection systems using reinforcement learning. Journal of Information Security and Applications, 61, 102923.
22. Sun, P., Liu, P., Li, Q., Liu, C., Lu, X., Hao, R., & Chen, J. (2020). DL-IDS: Extracting features using CNN-LSTM hybrid network for intrusion detection system. Security and communication networks, 2020, 1-11.
23. Zou, J., Han, Y., & So, S. S. (2009). Overview of artificial neural networks. Artificial neural networks: methods and applications, 14-22.